MAP-I
Programa Doutoral em Informática
Formal Models of Computation

Unidade Curricular em Teoria e Fundamentos
Theory and Foundations
(UCTF)
UP & UM

June 2015

′

Abstract

This is a proposal for a UCTF ("Unidade Curricular em Teorias e Fundamentos") course in the context of the joint PhD program (Minho, Aveiro, Porto) in Informatics (Map-I). The members of the team responsible for the proposal are lecturers from the University of Porto and University of Minho.

1 Lecturing Team

Sabine Broda, Nelma Moreira, Rogério Reis, José Carlos Espírito Santo

2 Course Description

2.1 Subject and Context

Models of computation are the foundations of Computer Science. Among them automata and lambda-calculi are most prominent both for theoretical and practical research.

Recent areas of application of the Automata Theory are computational linguistics, bioinformatics, specification and verification of reactive systems, structured and semi-structured databases, network security, etc. Intensively studied during mid XX's, in the last decade the exploration of specific new models and applications have at the same time stimulated a variety of new theoretical studies and formalizations. In this course we will consider regular languages and its extensions to infinite words, regular trees, quantitative systems, and regular relations. It is important to note that many algorithms and results for these models reduce or extend similar ones for classical finite automata. A special attention will be given to descriptional complexity issues, i.e. complexity measures, relative conciseness, complexity of language operations and of conversions between representations. These studies are motivated by the need to have good estimates of the amount of resources required to manipulate those representations. This is crucial in new applied areas where automata and other formal models are used, for instance for security certificates in formal verification systems. In general, having smaller objects improves our control on software, which may become shorter, more efficient and certifiable.

The lambda-calculus is a formalism originally introduced in the 1930's for the study of the foundations of mathematics, but soon after realized to provide also a universal model of computation, equivalent to Turing machines, appropriate for the study of computability. The lambda-calculus was rediscovered by computer scientists in the 1960's as the prototype of a functional programming language and as a platform for the programming languages semantics. In addition, the typed variants of the lambda-calculus exhibit a deep connection with proof systems, known as the Curry-Howard isomorphism. This correspondence provides a logical foundation for functional programming languages and their type systems, and a paradigm of correct programming.

The first part of the course reviews the basics of finite automata over finite words with emphasis on operations and conversions between equivalent language representations and their succinctness; descriptional complexity issues; and introduction to average case complexity. We also introduce weighted automata and transducers, and see how the standard finite automata algorithms are adapted.

The second part considers automata over infinite words and trees, including alternating automata, and briefly presents the main results regarding conversions and decidability problems, and its relation with logics with a particular emphasis on temporal logic.

The third part focuses on some recent applications of automata-theoretical approaches in different areas. Since the seminal Büchi theorem relating finite automata with monadic second order logic, automata have been successfully applied in many different logical contexts and specially with modal and temporal logics. These logics are particularly adequate for automatic verification of reactive systems. Weighted automata approaches were successfully applied in speech recognition and image processing to deal with probabilistic reasoning. Tree automata and finite automata are the theoretical bases for XML technologies such as schema, transformations and query languages. In Error Control Coding, transducers can be used to model various combinations of errors. In particular, many algorithms to test satisfaction and maximality of various error-detecting properties can be implemented based on state of the art transducer algorithms.

In the fourth part we will first introduce the basic notions and results about untyped lambda-calculus, show why it provides a universal model of computation, and consider abstract machines for the execution of lambda-terms. Later we will study typed lambda-calculi, survey their connections to proof systems through the Curry-Howard isomorphism. Finally we illustrate the use of abstract machines in the study of the semantics of functional programming languages, and the use of the Curry-Howard isomorphism in the correct development of programs from their specifications.

ACM Computing Classification System subjects covered:

- /Theory of Computation/Models of Computation/Computability/Lambda-Calculus
- /Theory of Computation/Models of Computation/Computability/Abstract machines/
- /Theory of Computation/Formal Languages and Automata Theory/Regular languages
- /Theory of Computation/Formal Languages and Automata Theory/Automata extensions/Transducer
- /Theory of Computation/Formal Languages and Automata Theory/Automata extensions/Quantitative Transducers
- /Theory of Computational/Logic/Verification by model checking

Similar courses There are several international institutions which provide courses on the topics cover by this course. One degree were there is a specialization in these topics is the Parisian Master of Research in Computer Science (MPRI) (`https://wikimpri.dptinfo.ens-cachan.fr/doku.phpMRPI`).

2.2 Objectives

This UCTF aims

- to present some recent research work in descriptional complexity of regular languages and to consider some open problems.
- to present some other automata models and see how the classical theory extends to them.
- to present recent applications of automata theory to other computer science areas, such as specification and verification of reactive systems and codes.
- to show the central place of the lambda-calculus in the theory of computation (as a universal model) and the theory of programming (in program semantics and correctness).

2.3 Learning Outcomes

- To understand the several representations for regular languages and operations, and their relative descriptional and computational complexity.
- To understand the several types of automata (alternating, weighted, timed, Büchi, over trees,etc) and how known algorithms for finite automata can be adapted.
- To understand the basic concepts of transducers and their applications.
- To understand the use of generating functions and analytic combinatorics for studying average-case analysis.
- To understand the several connections between logics and automata, specially monadic, temporal and modal logics.
- To understand the diversity of applications of automata and the gain of having common algorithms to apply to very different areas.
- To understand untyped lambda-calculus as a universal model of computation.
- To understand the role of lambda-calculus in the semantics of functional languages.
- To understand how the connection between typed lambda-calculi and proof systems provides a paradigm of correct program development.

2.4 Syllabus

- Part I: Automata on finite words (10h)
  1. Regular languages and their representations: regular expressions, deterministic finite automata (DFA), non-deterministic finite automata (NFA)
  2. Descriptional measures and operational complexities
  3. Enumeration and random generation of some classes of FAs
  4. Conversions between equivalent language representations and their succinctness
  5. Introduction to average case complexity based on analytic combinatorics: generating functions and analytic functions
  6. Alternating automata
  7. Weighted automata (automata with multiplicities)
  8. Transducers
- Part II: Automata on infinite words and trees (4h)

1. Büchi and Müller automata
2. Alternating automata
3. Timed automata
4. Tree automata
5. Automata and monadic logics

- Part III: Automata Applications (6h)

1. Verification of reactive systems
   (a) Temporal logics
   (b) Automata-theoretic approach to decidability
   (c) Model checking
2. (a) Tree languages and schema
   (b) XML language containment
3. Error Control Coding
   (a) code properties
   (b) satisfaction, maximality and unique decodability

- Part IV: Models of Functional Programming (8h)

1. Untyped lambda-calculus
2. Lambda-calculus as universal model of computation
3. Abstract machines
4. Types and Curry-Howard isomorphism
5. Applications
   (a) operational semantics of programming language
   (b) program verification and correctness

2.5 Student Assessment

- Examinations
- Research assignments, which may include a talk given on a suggested paper, or practical assignments

2.6 Recommended Bibliography

References

[Bar13] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, (2nd edition) 2013.

[DKV09] Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. Monographs In Theoretical Computer Science. An Eatcs Series. Springer, 2009.

[DS12] Deepak D'Souza and Priti Shankar, editors. *Modern Applications of Automata Theory*. Iisc Research Monographs Series. World Scientific, 2012.

[EMVM06] Z. Esik, C. Martín-Vide, and V. Mitrana, editors. *Recent Advances in Formal Languages and Appplications*. Springer-Verlag, 2006.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*. Springer-Verlag, 2002.

[HK11] Markus Holzer and Martin Kutrib. *Scientific Applications of Language Methods*, chapter Descriptional Complexity — An Introductory Survey, pages 1–58. Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory. Imperial College Press, carlos martín-vide edition, 2011.

[HS86] J.R. Hindley and J.P. Seldin. *Introduction to Combinators and [lambda]-calculus*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1986.

[Klu06] W. Kluge. *Abstract Computing Machines: A Lambda Calculus Perspective*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2006.

[KN01] B. Khoussainov and Anil Nerode. *Automata Theory and its Applications*. BirkhÄuser, 2001.

[RS97] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of FOrmal LAnguages*. Springer, 1997.

[Sak09] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[Sha08] Jeffrey Shallit. *A Second Course in Formal Languages and Automata Theory*. Cambridge University Press, 2008.

[SU06] M.H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2006.

[Wan12] Jiacun Wang, editor. *Handbook of Finite State Based Models and Applications*. Discrete Mathematics And Its Applications. Chapman And Hall/Crc Press, 2012.

3 Lecturing Team

The proponents of this course have worked actively in the past few years, on topics that are directly related to the subjects covered by this course, as detailed below.

In the context of this course, we also would like to have the opportunity to invite internationally recognized researchers such as Markus Holzer (Universität Giessen, Germany), who mainly research on computational and descriptional complexity of formal languages, or Jacques Sakarovitch (ENST, Paris), one of his topics of research is automata with multiplicities.

- Sabine Broda, has worked on Mathematical Logic. Her current research interests include automata theory and formal languages; descriptional complexity; and formal verification. She has been publishing (as well as refereeing) regularly in international journals and conferences on these areas.

- Nelma Moreira has worked in Automata Theory, Modal Logics, Verification and Logic Programming. She has several publications on international journals and conferences concerning descriptional complexity of regular languages, succinct conversions between equivalent models of regular languages and, average-case complexity based on analytic combinatorics. She has been member of program committees of several international conferences. She is member of IFIP Working Group 1.2 since 2013 and member of the steering committee of NCMA (Workshop on Non-Classical Models of Automata).

- Rogério Reis, Phd in Automata Theory, his main research interests are in the area of formal languages and automata theory; descriptional complexity; enumerative and analytic combinatorics; and cryptography. He has several scientific publications in international journals and conferences. He has been member of program committees of several international conferences. He is member of IFIP Working Group 1.2 since 2013 and member of the steering committee of DCFS (International Workshop Descriptional Complexity of Formal Systems).

- José Carlos Espírito Santo works in the areas of lambda-calculus (strong normalisation), proof theory (sequent calculus, Curry-Howard isomorphism, proof search) and type theory (intersection types). He publishes regularly in journals and international conferences on these areas.

One proponent was the principal investigator and two others were team members of the project CANTE (Descriptional and computational complexity of formal languages, PTDC/EIA-CCO/101904/2008) 2010-2013 which aimed to obtain new, concrete characterizations of formal language representations, both from the descriptional as well as computational complexity perspective, leading to efficient manipulation methods with application to the construction of automata based certificates of program properties, and algebraic and coalgebraic methods for proof systems based on Kleene algebras. Two proponents were team members of the project AVIACC (Analysis and Verification of Critical Concurrent Programs, PTDC/EIA-CCO/117590/2010) during which they pursued work on the use of Kleene algebras and automata as devices for studying properties of concurrent systems. The general goal of this work was the definition of a common framework for modeling and analyzing different kinds of concurrent systems, including the design of adequate and feasible decision algorithms.

All proponents were team members of the project RESCUE (Reliable and Safe Code Execution for Embedded Systems, FCT/PTDC/EIA/65862/2006), where they aimed to apply automata-theoretical techniques to verification in the context of PPC (Proof-Carry Code).

Some of them were also members of sites in the European network TYPES, a Coordination Action on Type Theory, funded by European Union.