

Course proposal for ESSLLI 2016

Bolzano, 15–26 August, 2016

Title: Logics on words and trees with data

Area: Logic & Computation

Level: Introductory

Lecturers:

Diego Figueira

Affiliation: CNRS, LaBRI, France

Email: diego.figueira@labri.fr

Homepage: <http://homepages.inf.ed.ac.uk/dfigueir/>

Ranko Lazić

Affiliation: University of Warwick, UK

Email: r.s.lazic@warwick.ac.uk

Homepage: http://www2.warwick.ac.uk/fac/sci/dcs/people/Ranko_Lazic/

Abstract.

This course will present several results linking logics for semistructured data (such as XML) with counter systems. The course will focus on two well-studied data models: data words, and data trees. These are words and trees whose every element carries a label from a finite alphabet and a data value from an infinite domain; indeed these are standard abstractions for semi-structured documents. The focus is on the complexity and decidability of reasoning on these structures. The plan is to show three groups of logics with very differing expressive power and capabilities, in order to give an overall idea of the state of the art and different powerful techniques for proving decidability in the area. These logics are divided into: first-order logics, temporal logics, and path logics.

The course material should be useful to anyone with an interest in query languages for semi-structured data, counter systems or more generally on verification of infinite-state systems. This course has some technically demanding parts, and should appeal mainly to an audience from Logic, Verification and Theoretical Computer Science.

Motivation and description

Words and trees are amongst the most studied structures in computer science. In this course, we focus on words and trees that can contain elements from some infinite alphabet,

like for example the set of integers, or the set of words over the alphabet $\{a, b\}$. These kind of structures are relevant to many areas.

In software verification, one may need to decide statically whether a program satisfies some given specification; and the necessity of dealing with infinite alphabets can arise from different angles. For example, in the presence of concurrency, we have an unbounded number of processes running, each one with its process identification, and we must verify properties specifying the interplay between these processes. Further, procedures may take parameters as input, and they can hence exchange data from some unbounded domain. Infinite alphabets can also emerge as a consequence of the use of recursive procedure calls, communication through FIFO channels, etc.

On the other hand, in a database context, infinite alphabets are a common occurrence. Let us dwell on static analysis tasks on XML documents and its query languages. In this context, there are several pertinent problems serving static analysis. For example, there is the problem of coherence: is there a document in which a given query returns a non-empty result? The problem of inclusion: is it true that for any document, the result answered by one given query is contained in the result of another? And the problem of equivalence: do two given queries always return the same answers? In the database context, these questions are at the core of many static analysis tasks. For example, by answering the coherence problem one can decide whether the computation of a query on a database can be avoided because the query contains a contradiction; and by the equivalence problem if one query can be safely replaced by a simpler one. All these queries recurrently need to specify properties concerning not only the labels of the nodes, but also the actual data contained in the attributes.

Still in the context of databases, we can also regard logics that express data properties as specification languages. In verification of database-driven systems, we are provided with the specification of a system that interacts with a database, and we need to check whether it is possible to reach a state in which the database has some undesired property. In order to model the specification of the system as well as the property of the database—for example through an automaton or a logical formula—we typically need to take into account values from infinite domains.

Therefore, the study of formalisms to reason with words and trees that can carry elements from some infinite domain is relevant to all the aforementioned areas, and possibly more.

Structure of the course and tentative schedule

The course will be structured into five lessons of about 90 minutes each.

Below is a tentative schedule of the covered subjects.

Day 1. (a) Data words, data trees. Motivations: XML, temporal databases, concurrency

- (b) Counter systems. Minsky machine, (inflationary) counter machine. Reachability, control-state reachability, coverability problems.

Part I: data words

Day 2. First-order logic on data words

- (a) Undecidability of $\text{sat-FO}(<, \sim)$
- (b) $\text{FO}^2(<, +1, \sim) \equiv \text{reach}(\text{VAS})$
- (c) Data automata

Day 3. Temporal logics

- (a) LTL with freeze, alternating register automata, decidability
- (b) Lower bounds: $\text{LTL}(F)$ non-PR, $\text{LTL}(F, F^{-1})$ undecidability
- (c) A restriction of LTL: LRV (adapted to data words) $\text{sat}(\text{LRV}) \equiv \text{reach}(\text{VAS})$, $\text{sat}(\text{forward-LRV}) \equiv \text{cover}(\text{VAS})$
- (d) Model checking LTL over one-counter automata

Part II: data trees

Day 4. Path logics

- (a) XPath. Undecidability of sat-XPath .
- (b) Decidable fragments: downward, forward, vertical, reflexive-transitive.
- (c) ATRA1, decidability, connection with XPath.

Day 5. Other formalisms on data trees

- (a) $\text{FO}^2(<, +1) \equiv \text{reach}(\text{BVASS})$, $\text{FO}^2(+1)$ decidable
- (b) μ -calculus with registers
- (c) Data patterns

Requirements

This course will use some notions from the proposed course on well-quasi-orderings “Algorithmic Aspects of WQO Theory”, and it is suggested as a follow-up course on the second week of ESSLLI. However, the two courses are independent, and following the first course is not a pre-requisite to take part of the second.

Expected external funding

We expect to be able to obtain partial or full funding from our host institutions.

References

1. L. Segoufin: Automata and Logics for Words and Trees over an Infinite Alphabet. CSL 2006: 41–57.
2. M. Bojańczyk, C. David, A. Muscholl, Th. Schwentick and L. Segoufin: Two-variable logic on words with data. Trans. on Computational Logic (ToCL) 12(4), 2011.
3. S. Demri and R. Lazić: LTL with the Freeze Quantifier and Register Automata. Trans. on Computational Logic (ToCL) 10(3), 2009.
4. S. Demri, R. Lazić and A. Sangnier: Model checking memoryful linear-time logics over one-counter automata. Theoretical Computer Science 411(22–24): 2298–2316, 2010.
5. M. Bojanczyk, A. Muscholl, Th. Schwentick and L. Segoufin: Two-variable logic on data trees and XML reasoning. J. ACM 56(3), 2009.
6. D. Figueira: Reasoning on words and trees with data. Ph.D. Thesis, Laboratoire Spécification et Vérification, ENS Cachan, France, 2010.
7. M. Jurdziński and R. Lazić: Alternating automata on data trees and XPath satisfiability. Trans. on Computational Logic (ToCL) 12(3), 2011.