Logic and regular cost functions

Thomas Colcombet
IRIF, CNRS and Université Paris Diderot – Paris 7, France
thomas.colcombet@irif.fr

***Abstract*—Regular cost functions offer a toolbox for automatically solving problems of existence of bounds, in a way similar to the theory of regular languages. More precisely, it allows to test the existence of bounds for quantities that can be defined in cost monadic second-order logic (a quantitative variant of monadic second-order logic) with inputs that range over finite words, infinite words, finite trees, and (sometimes) infinite trees.**

Though the initial results date from the works of Hashiguchi in the early eighties, it is during the last decade that the theory took its current shape and that many new results and applications have been established.

In this tutorial, two connections linking logic with the theory of regular cost functions will be described. The first connection is a proof of a result of Blumensath, Otto and Weyer stating that it is decidable whether the fixpoint of a monadic second-order formula is reached within a bounded number of iterations over the class of infinite trees. The second connection is how non-standard models (and more precisely non-standard analysis) give rise to a unification of the theory of regular cost functions with the one of regular languages.

Overview

A very narrow perspective on the theory of regular languages can be stated as follows:

> **Regular languages**
>
> The theory of *regular languages* offers a toolbox of notions and algorithms for answering questions of the form:
>
> Do all inputs satisfy a given property?
>
> in which
>
> *inputs* may range over words, infinite words, trees, infinite trees, or graphs of given bounded tree-width or clique-width, and
>
> *properties* are expressed in some logical formalism, the standard being monadic second-order logic (MSO for short), but other possibilities also exist such as fixpoint logics (μ-calculus), and so on. In this view, automata provide simply a convenient syntax for fragments (with good properties) of these logics.

Indeed, this is a very narrow presentation since most results in language theory do not directly fall in this description. Nevertheless, it is a fair description of why regular languages play a central role in "algorithmic model-theory", which in a broad sense consists of the methods used for deciding the satisfiability of logic formulae and checking of models.

The author is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No.670624), and by the DeLTA ANR project (ANR-16-CE40-0007).

978-1-5090-3018-7/17/$31.00

Taking the same (narrow) point of view, we can present the theory of regular cost functions in a similar way:

> **Regular cost functions**
>
> The theory of *regular cost functions* offers a toolbox of notions and algorithms for answering questions of the form:
>
> Is some quantity bounded over all inputs?
>
> in which
>
> *inputs* may range over words, infinite words, trees, infinite trees, or graphs of given bounded tree- or clique-width, and
>
> *quantities* in $\mathbb{N} \cup \{\infty\}$ are described in some logical formalism, the standard being cost monadic second-order logic (cost MSO for short), but other possibilities exist such as fixpoint logics (*cost μ-calculus*), and so on.

Usually, describing the theory of regular cost functions in more detail requires to enter the technicalities of the different notions of acceptors (B-automata, S-automata, $\sharp$-monoids, cost games, and so on) and the involved constructions relating them. In this tutorial we deliberately ignore this (essential) aspect, and instead underline some further connections between logic and regular cost functions. We will in particular emphasize on the two following points:

A. we will explain how regular cost functions can be seen as a form of regular languages thanks to the use of non-standard analysis [15], and,
B. as an application, we will show how to decide whether the fixpoint of some MSO formula is reached in a bounded number of iterations over all (infinite) trees or not [3], a result due to Blumensath, Otto and Weyer.

We will not touch these aspects in this short abstract, but rather simply give some high levels ideas of what are regular cost functions. This begins with some key definitions in the following section. We then describe some results related to logic that can be solved using regular cost functions. We conclude with some historical pointers.

Cost monadic second-order logic and regular cost functions

Let us recall that *monadic second-order logic* is the extension of first-order logic with *set quantifiers* ($\exists X \ldots$, in which *set variables* are traditionally in capital letters) and a *membership predicate* ($y \in X$) testing whether a first-order variable belongs to a set variable.

As an example, monadic second-order logic can express the reachability between vertices in a directed graph. Indeed, there is a path in a digraph from vertex u to vertex v if and only if

> "all sets X that contain u and are closed under the edge relation (i.e. for all edges such that the source belongs to X, the target also does) also contain v."

This can be phrased in plain formulae as follows:

$$\mathtt{path}(u,v) := \forall X\ (u \in X \wedge \mathtt{closed}(X)) \to v \in X\ ,$$
$$\mathtt{closed}(X) := \forall x \forall y\ (\mathtt{edge}(x,y) \wedge x \in X) \to y \in X\ .$$

Cost monadic second-order logic extends monadic second-order logic with a single construct:

$$|X| \leq n\ ,$$

in which X is a set variable and n is the unique representent of *number variables*, and ranges over $\mathbb{N}$. The semantics is as expected: it means "the cardinality of the set X is at most n". The important thing is that this construct is only allowed to appear *positively* in a formula, i.e., *under an even number of negations*.

A formula of cost monadic second-order logic φ naturally associates to each input structure $\mathfrak{A}$ the quantity

$$[\![\varphi]\!](\mathfrak{A}) = \inf\{i \in \mathbb{N}\ \mid\ \mathfrak{A}, n = i \models \varphi\} \in \mathbb{N} \cup \{\infty\}\ ,$$

in which $\mathfrak{A}, n = i \models \varphi$ signifies that $\mathfrak{A}$ is a model of φ when n takes the value i.

Our first example describes the distance between two vertices in a directed graph. Indeed, there is a path of length at most n from a vertex u to a vertex v if and only if:

> "There exists a set of nodes Z of size at most n such that in the digraph with edges restricted to the ones ending in Z there is a path from u to v."

In formulae, we obtain something like:

$$\begin{aligned}\mathtt{dist}(u,v) &:= \exists Z\ |Z| \leq n \\ &\qquad \wedge \forall X\ (u \in X \wedge \mathtt{closed}(X,Z)) \to v \in X\ , \\ \mathtt{closed}(X,Z) &:= \forall x \forall y \\ &\qquad (\mathtt{edge}(x,y) \wedge x \in X \wedge y \in Z) \to y \in X\ .\end{aligned}$$

For a directed graph $\mathfrak{G}$, then $\mathfrak{G}, n = i \models \mathtt{dist}(u,v)$ if and only if there exists a path of length at most n from u to v in $\mathfrak{G}$. This means that $[\![\mathtt{dist}(u,v)]\!](\mathfrak{G}) \in \mathbb{N} \cup \{\infty\}$ is the distance between u and v, or ∞ if there is no path between these two nodes.

As done classically since the seminal works of Elgot and Büchi, (finite) words can be identified as (finite) chains, i.e., finite relational structure over the signature consisting of a binary symbol $<$ interpreted as the linear order of positions, and one monadic predicate for each letter interpreted as the set of positions that carry this letter. Under this interpretation it is possible to express in cost monadic second-order logic functions from words to $\mathbb{N} \cup \{\infty\}$. For instance, we can consider the function F wich to a word over the alphabet $\{a, b, c\}$ associates the least n such as:

> "There are at most n occurrences of the letter a, or there exists a maximal interval of consecutive positions that does not contain a letter a, that contains at most n occurrences of the letter b."

As mentioned, the central motivation of the theory of regular cost functions is to be able to decide boundedness problems. For f mapping structures (of suitable signature) to $\mathbb{N} \cup \{\infty\}$, deciding the *boundedness of f over a class $\mathcal{C}$ of structures* is the problem of deciding whether there exists an integer m such that $f(\mathfrak{A}) \leq m$ for all structures $\mathfrak{A} \in \mathcal{C}$.

For instance, in the above example, the function computed is not bounded since we can use, for instance, the words

$$u_n = b^n(ab^n)^n\ , \qquad \text{for all } n \in \mathbb{N},$$

as witnesses of unboundedness. Indeed, $F(u_n) = n$ for all $n \in \mathbb{N}$.

One essential result in the theory of regular cost functions can be stated as follows:

Theorem 1: Given a formula of cost monadic second-order logic φ, it is decidable whether $[\![\varphi]\!]$ is bounded

- over finite words [14], [16],
- over words of length ω [34],
- over finite trees [23], and as a consequence over graphs of bounded tree-width or clique-width,
- over infinite trees for formulae of *weak*[1] cost monadic second-order logic [8], or weak cost monadic second-order logic extended with bounded monadic fixpoints [2].

One can see in this list of results the successive generalizations to regular cost functions of the classical results of Büchi and Elgot [11], [26], Büchi [12], Thatcher and Wright [44], Courcelle [25] and partly Rabin [42]. Indeed, what would be the equivalent of the full result of Rabin is still beyond reach and is the big open question in this area:

Open problem 1: Decide whether a formula of cost monadic second-order logic computes a function bounded over infinite trees.

One may wonder at this stage why we refer to regular cost functions[2] and not simply "regular functions" (appart from the fact that it would certainly be a very ambiguous terminology). This comes from the following definitions: consider two maps f, g from some set E (think words, or trees, ... or more generally inputs) to $\mathbb{N} \cup \{\infty\}$, then f is *equivalent to* g if for all $X \subseteq E$, $f|_X$ is bounded[3] if and only if $g|_X$ is bounded. This is denoted $f \approx g$. A *cost function* is an equivalence relation for this equivalence $\approx$. A *regular cost function* is a cost function that contains a function computed by a cost monadic second-order formula.

The constructions performed within the theory of regular cost functions do not preserve in general the computed function exactly, but only up to $\approx$. For instance, the proofs of

[1] In which set variables do only range over finite sets
[2] To not confuse with cost register automata.
[3] meaning by some value from $\mathbb{N}$.

Theorem 1 involve many statements of the form "a formula in a given family can be translated into an equivalent automaton of suitable form". What is meant by "equivalent" in these statements is that the cost function computed is preserved, but not necessarily the exact function. This is why the notion of cost function is the unifying concept in this approach.

SOME LOGIC RELATED QUESTIONS SOLVABLE USING REGULAR COST FUNCTIONS

As mentioned above, the following result will be presented during the talk:

Theorem 2 (Blumensath, Otto and Weyer [3]): Given a formula $\varphi(x, X)$ of monadic second-order logic, positive[4] in X, it is decidable whether there exists an integer n such that over all (infinite) trees t, the least fixpoint of φ is reached within n iterations.

This result is a typical application of regular cost functions. Let us mention here some other related problems in logics that can be solved or at least attacked using the theory of regular cost functions. These three questions have been chosen to be different in style.

Σ_2-separation: A Σ_2-*formula* is a first-order formula such that no existential quantifier appears below a universal one. The Σ_2-*separation problem* takes two regular languages of words K and L, and consists in deciding whether there exists a Σ_2-formula that contains K but does not intersect L.
This problem has been shown decidable recently [41]. It is in fact very close to the relative inclusion star-height problem [28], [29], [32], and the use of regular cost functions offer much simpler proofs.

Quantitative Church synthesis problem: The *Church synthesis problem* takes as input a regular language L of words of length ω (i.e., defined by a monadic second-order formula), and asks whether in a game where two *players I* and *II* alternatively choose letters, player I can guarantee that the infinite play resulting from the game belongs to L. In their seminal work, Büchi and Landweber have shown this problem decidable [13]. Rabinovich and Velner [45] left open a similar question: in the *quantitative Church synthesis problem* player I can furthermore change up to n letters at the end of the game. The question is whether there exists some n such that player I can guarantee the play to belong to L. The answer is positive, and a direct application of regular cost functions.

Rabin-Mostowski hierarchy problem: the *Rabin-Mostowski index* of a property is (essentially) the number of alternations of least and greatest fixpoints necessary for describing this property in the μ-calculus [37]. This index is known to form a strict hierarchy for general μ-calculus formulae [10], [35], [1], as well as for fragments, and in particular for the "disjunctive fragment" that corresponds to non-deterministic automata [39], [38]. Computing this index is known as the *Rabin-Mostowski hierarchy problem*, and is open in both cases (only some particular cases were known to be decidable [39], [40]). However, in the disjunctive case, it is known to be reducible to the boundedness question for cost monadic second-order logic over infinite trees [22], i.e., to *Open problem 1*. Though the general problem remains undecidable, the approach using regular cost functions has helped showing other cases decidable [24].

[4]A predicate of the form $y \in X$ for some first order variable y is only allowed to appear positively in the formula.

REFERENCES

Regular cost functions arose from the seminal works of Hashiguchi, Leung and Simon [27], [36], [43], initiated by Hashiguchi for solving the *star height problem*: the problem of deciding, given a regular language, whether it can be represented by a regular expressions of fixed nesting of Kleene stars [28]. Kirsten gave a simplified proof of this result [30], [31], introducing essential new forms of automata along the way. This problem is also known to be decidable over finite trees [21]. Independently, Bojańcyk introduced the logic MSO+U[5] [4]. For solving more interesting fragments of this logic, Bojańczyk and Colcombet introduced new forms of automata (which in some precise sense generalize the ones of Kirsten) [6] (in the end this logic has been shown undecidable [7]). The notion of regular cost function is then isolated as well as the algebraic notion of "♯-monoids" and cost monadic second-order logic in [14], [16]. There, the boundedness (and more general domination) problem is solved over finite words. Some automata results were also later simplified in [5], [17]. These results are extended to trees in [23], and to infinite words in [34]. For infinite trees, only partial results are known [8], [9], [2]. Efforts were also devoted to the characterization (in the spirit of Schützenberger) of subclasses of regular cost functions [19], [33], [20]. Applications to games have been investigated in [18].

REFERENCES

[1] A. Arnold. The μ-calculus alternation-depth hierarchy is strict on binary trees. *Informatique Théorique et Applications*, 33(4/5):329–340, 1999.

[2] Achim Blumensath, Thomas Colcombet, Denis Kuperberg, Michael Vanden Boom, and Paweł Parys. Two-way cost automata and cost logics over infinite trees. In *CSL-LICS 2014: 23rd EACSL Annual Conference on Computer Science Logic and 29th Annual ACM/IEEE Symposium on Logic in Computer Science*, page 16, 2014.

[3] Achim Blumensath, Martin Otto, and Mark Weyer. Decidability results for the boundedness problem. *Logical Methods in Computer Science*, 10(3), 2014.

[4] Mikołaj Bojańczyk. A bounding quantifier. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *CSL 2004: Computer Science Logic, 18th International Workshop*, volume 3210 of *Lecture Notes in Comput. Sci.*, pages 41–55. Springer, 2004.

[5] Mikołaj Bojańczyk. Star height via games. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 214–219, 2015.

[6] Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω-regularity. In *LICS 06: 21th IEEE Symposium on Logic in Computer Science*, pages 285–296. IEEE Computer Society, 2006.

[5]Monadic second-order logic enhanced with a new predicate expressing that there exists sets of arbitrarily large size satisfying a given formula. This is only meaningful for infinite models: words of length ω is the yardstick.

[7] Mikołaj Bojańczyk, Pawel Parys, and Szymon Toruńczyk. The MSO+U theory of (n, <) is undecidable. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 21:1–21:8, 2016.

[8] Michael Vanden Boom. Weak cost monadic logic over infinite trees. In Filip Murlak and Piotr Sankowski, editors, *MFCS 2011: Mathematical Foundations of Computer Science*, volume 6907 of *Lecture Notes in Comput. Sci.*, pages 580–591. Springer, 2011.

[9] Michael Vanden Boom and Denis Kuperberg. Quasi-weak cost automata: A new variant of weakness. In Supratik Chakraborty and Amit Kumar, editors, *FSTTCS 2011: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 13 of *LIPIcs*, pages 66–77. Schloss Dagstuhl, 2011.

[10] J.C. Bradfield. The modal μ-calculus alternation hierarchy is strict. *Theoret. Comput. Sci.*, 195(2):133–153, 1998.

[11] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.

[12] J. Richard Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr .)*, pages 1–11. Stanford Univ. Press, Stanford, Calif., 1962.

[13] J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.*, 138:295–311, 1969.

[14] Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP 2009 (2): Automata, Languages and Programming, 36th International Colloquium*, volume 5556 of *Lecture Notes in Comput. Sci.*, pages 139–150. Springer, 2009.

[15] Thomas Colcombet. Cost functions with several orders of magnitudes and the use of relative internal set theory. In *LICS*, page 123, 2013.

[16] Thomas Colcombet. Regular cost functions, part I: logic and algebra over words. *Log. Methods Comput. Sci.*, page 47, 2013.

[17] Thomas Colcombet and Nathanaël Fijalkow. The Bridge Between Regular Cost Functions and Omega-Regular Languages. In Yuval Rabani Ioannis Chatzigiannakis, Michael Mitzenmacher and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 126:1–126:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[18] Thomas Colcombet and Stefan Göller. Games with bound guess actions. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 257–266, 2016.

[19] Thomas Colcombet, Denis Kuperberg, and Sylvain Lombardy. Regular temporal cost functions. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *ICALP 2010 (2): Automata, Languages and Programming, 37th International Colloquium*, volume 6199 of *Lecture Notes in Comput. Sci.*, pages 563–574. Springer, 2010.

[20] Thomas Colcombet, Denis Kuperberg, Amaldev Manuel, and Szymon Torunczyk. Cost Functions Definable by Min/Max Automata. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[21] Thomas Colcombet and Christof Löding. The nesting-depth of disjunctive μ-calculus for tree languages and the limitedness problem. In Michael Kaminski and Simone Martini, editors, *CSL 2008: Computer Science Logic, 22nd International Workshop*, volume 5213 of *Lecture Notes in Comput. Sci.*, pages 416–430. Springer, 2008.

[22] Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008 (2): Automata, Languages and Programming, 35th International Colloquium*, volume 5126 of *Lecture Notes in Comput. Sci.*, pages 398–409. Springer, 2008.

[23] Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS 2010: 5th Annual IEEE Symposium on Logic in Computer Science*, pages 70–79. IEEE Computer Society, 2010.

[24] Thomas Colcombet, Christof Löding, Denis Kuperberg, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, pages 215–230, 2013.

[25] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

[26] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, 1961.

[27] Kosaburo Hashiguchi. Limitedness theorem on finite automata with distance functions. *J. Comput. Syst. Sci.*, 24(2):233–244, 1982.

[28] Kosaburo Hashiguchi. Relative star height, star height and finite automata with distance functions. In Jean-Eric Pin, editor, *Formal Properties of Finite Automata and Applications*, volume 386, pages 74–88, 1988.

[29] Kosaburo Hashiguchi. Algorithms for determining relative inclusion star height and inclusion star height. *Theoret. Comput. Sci.*, 91(1):85–100, 1991.

[30] Daniel Kirsten. Distance desert automata and the star height one problem. In Igor Walukiewicz, editor, *FoSSaCS 2004: Foundations of Software Science and Computation Structures, 7th International Conference*, volume 2987 of *Lecture Notes in Comput. Sci.*, pages 257–272. Springer, 2004.

[31] Daniel Kirsten. Distance desert automata and the star height problem. *RAIRO - Theor. Inf. Appl.*, 3(39):455–509, 2005.

[32] Daniel Kirsten. On the complexity of the relative inclusion star height problem. *Advances in Computer Science and Engineering*, 5(2):173–211, 2010.

[33] Denis Kuperberg. Linear temporal logic for regular cost functions. In Thomas Schwentick and Christoph Dürr, editors, *STACS 2011: 28th International Symposium on Theoretical Aspects of Computer Science*, volume 9 of *LIPIcs*, pages 627–636. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.

[34] Denis Kuperberg and Michael Vanden Boom. On the expressive power of cost logics over infinite words. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP 2012 (2): Automata, Languages, and Programming - 39th International Colloquium*, volume 7392 of *Lecture Notes in Comput. Sci.*, pages 287–298. Springer, 2012.

[35] Giacomo Lenzi. A hierarchy theorem for the μ-calculus. In Friedhelm Meyer auf der Heide and Burkhard Monien, editors, *ICALP 1996: Automata, Languages and Programming, 23rd International Colloquium*, volume 1099 of *Lecture Notes in Comput. Sci.*, pages 87–97. Springer, 1996.

[36] Hing Leung. *An Algebraic Method for Solving Decision Problems in Finite Automata Theory*. PhD thesis, Pennsylvania State University, Department of Computer Science, 1987.

[37] Andrzej W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *Symposium on Computation Theory*, pages 157–168, 1984.

[38] Damian Niwiński. On fixed-point clones. In Laurent Kott, editor, *ICALP 1986: Automata, Languages and Programming, 13th International Colloquium*, volume 226 of *Lecture Notes in Comput. Sci.*, pages 464–473. Springer, 1986.

[39] Damian Niwinski and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

[40] Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput.*, 123:195–208, 2005.

[41] Thomas Place and Marc Zeitoun. Going higher in the first-order quantifier alternation hierarchy on words. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 342–353, 2014.

[42] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[43] Imre Simon. Recognizable sets with multiplicities in the tropical semiring. In Michal Chytil, Ladislav Janiga, and Václav Koubek, editors, *MFCS 1988: Mathematical Foundations of Computer Science*, volume 324 of *Lecture Notes in Comput. Sci.*, pages 107–120. Springer, 1988.

[44] James W. Thatcher and Jesse B. Wright. Generalized automata theory with an appliucation to a decision problem in second-order logic. *Math. Syst. Theory*, 2:57–81, 1968.

[45] Yaron Velner and Alexander Rabinovich. Church synthesis problem for noisy input. In Martin Hofmann, editor, *FoSSaCS 2011: Foundations of Software Science and Computational Structures - 14th International Conference*, volume 6604 of *Lecture Notes in Comput. Sci.*, pages 275–289. Springer, 2011.