B. Courcelle

Algebraic and Regular Trees

Publications du Département de Mathématiques de Lyon, 1985, fascicule 2B
« Compte rendu des journées infinitistes », , p. 91-95
<http://www.numdam.org/item?id=PDML_1985___2B_91_0>

ALGEBRAIC AND REGULAR TREES

by B. COURCELLE

The lecture was based on the following two published papers of which I reproduce below the introductions and lists of references.

Theoretical Computer Science 25 (1983) 95–169
North-Holland Publishing Company

FUNDAMENTAL PROPERTIES OF INFINITE TREES

Bruno COURCELLE
UER de Mathématiques et Informatique, Université Bordeaux-I, 33405 Talence, France

Introduction

Infinite trees naturally arise in mathematical investigations on the semantics of programming languages. They arise in essentially two ways: when one *unloops* or *unfolds* a program undefinitely. One obtains then either a *tree of execution paths* (infinite in general) in the case of a program written in an imperative language like FORTRAN or an *expression tree* in the case of a program written in an applicative language like LISP. In the latter case, the expression tree is usually infinite although its value can be finitely computed in each case; this is possible by the use of **if–then–else** as a base function (like the addition of integers) and *not* as a piece of control structure. Once again, the infiniteness of the tree corresponds to the infiniteness of the set of possible computations.

In both cases, the semantics of the program is completely defined by the associated tree. Hence two programs are equivalent if the associated trees are the same (the converse being not true). Roughly speaking, this allows to distinguish between the equivalence of programs which is only due to the control structure (loops, recursive calls, etc. . . .) from the equivalence which also depends on the properties of the domains of computation and the given 'base' functions on these domains.

It should be noted that these infinite trees are finitely defined. Hence we are lead to try to decide whether two infinite trees defined in some finitary way are equal.

Two types of infinite trees will be considered: the *regular trees* which are defined by unlooping FORTRAN-like program or flowcharts and the *algebraic trees* which are defined by unfolding recursive program schemes more or less derived from LISP programs.

We shall introduce operations on trees: the *first-order substitution* which corresponds (roughly) to the sequential composition of flowcharts (by the operator; of ALGOL) or to functional application (in the case of an applicative language). We shall also introduce the *second-order substitution* which corresponds to the replacement of a function symbol in an expression tree by some expression tree intended to denote the corresponding function.

Here is a brief survey of the content of the paper which is intended to be a synthesis of several aspects of infinite trees usually defined and studied separately for different purposes:

(1) *Topological* (i.e. metric) and *order-theoretical properties* of infinite trees are investigated in parallel in order to enlighten similarities and differences.

(2) *First-* and *second-order substitutions* are investigated in the two above frameworks.

(3) *Regular trees*, rational expressions defining them are studied. The concept of an iterative theory, due to C.C. Elgot, is one of the possible algebraic frameworks where to study infinite trees: the set of regular trees forms the free iterative theory. Regular trees also arise as most general first-order unifiers in a generalized sense.

(4) *Algebraic trees* play a similar role with respect to second-order substitutions as regular trees with respect to first-order ones. Their combinatorial properties are sufficiently complicated to yield an open problem equivalent to the DPDA equivalence problem.

References

[1] A. Arnold and M. Dauchet, Théorie des magmoïdes, *RAIRO Inform. Théor.* **12** (1978) 235–257.

[2] A. Arnold and M. Nivat, Metric interpretations of infinite trees and semantics of non deterministic recursive programs, *Theoret. Comput. Sci.* **11** (1980) 181–205.

[3] A. Arnold and M. Nivat, The metric space of infinite trees. Algebraic and topological properties, *Fund. Inform.* **III.4** (1980) 445–476.

[4] H. Bekič, Definable operations in general algebras, and the theory of automata and flowcharts, Unpublished work, IBM Laboratory, Vienna (1969).

[5] S. Bloom, All solutions of a system of recursion equations in infinite trees and other contraction theories, *J. Comput. System Sci.*, to appear.

[6] S. Bloom and C. Elgot, The existence and construction of free iterative theories, *J. Comput. System Sci.* **12** (1976) 305–318.

[7] S. Bloom, C. Elgot and J. Wright, Solutions of the iteration equation and extensions of the scalar iteration operation, *SIAM J. Comput.* **9** (1980) 25–45.

[8] S. Bloom, S. Ginali and J. Rutledge, Scalar and vector iteration, *J. Comput. System Sci.* **14** (1977) 251–256.

[9] S. Bloom and D. Patterson, Easy solutions are hard to find, *Proc. 6th Colloquium on Trees in Algebra and Programming*, Lecture Notes in Computer Science **112** (Springer, Berlin, 1981) 135–146.

[10] S. Bloom and R. Tindell, Compatible orderings on the metric theory of trees, *SIAM J. Comput.* **9** (1980) 683–691.

[11] S. Bloom and R. Tindell, Varieties of 'if-then-else', Submitted for publication (1981).

[12] R. Book, The undecidability of a word problem: on a conjecture of Strong, Maggiolo-Schettini and Rosen, *Inform. Process Lett.* **12** (1981) 121–122.

[13] P. Casteran, Structures de contrôle: définitions opérationnelles et algébriques, Thèse de 3ème cycle, University Paris-7 (1979).

[14] B. Courcelle, On jump-deterministic pushdown automata, *Math. Systems Theory* **11** (1977) 87–109.

[15] B. Courcelle, A representation of trees by languages, *Theoret. Comput. Sci.* **6** (1978) 255–279 and **7** (1978) 25–55.

[16] B. Courcelle, Frontiers of infinite trees, *RAIRO Inform. Théor.* **12** (1978) 319–337.

[17] B. Courcelle, Arbres infinis et systèmes d'équations, *RAIRO Inform. Théor.* **13** (1979) 31–48.

[18] B. Courcelle, Infinite trees in normal form and recursive equations having a unique solution, *Math. Systems Theory* **13** (1979) 131–180.

[19] B. Courcelle, An axiomatic approach to the Korenjak–Hopcroft algorithms, *Math. Systems Theory*, to appear.

[20] B. Courcelle, Work in preparation.

[21] B. Courcelle and P. Franchi-Zannettacci, On the equivalence problem for attribute systems, *Information and Control*, to appear.

[22] B. Courcelle and I. Guessarian, On some classes of interpretations, *J. Comput. System Sci.* **17** (1978) 388–413.

[23] B. Courcelle, G. Kahn and J. Vuillemin, Algorithmes d'équivalence et de réduction à des expressions minimales, dans une classe d'équations récursives simples, *Proc. 2nd International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science **14** (Springer, Berlin, 1974) 200–213.

[24] B. Courcelle and M. Nivat, Algebraic families of interpretations, *Proc. Annual Symposium on Foundations of Computer Science*, Houston, TX (1976) 137–146.

[25] B. Courcelle and M. Nivat, The algebraic semantics of recursive program schemes, in: *Mathematical Foundations of Computer Science '78*, Lecture Notes in Computer Science **64** (Springer, Berlin, 1978) 16–30.

[26] B. Courcelle and J.C. Raoult, Completions of ordered magmas, *Fund. Inform.* **III.1** (1980) 105–116.

[27] B. Courcelle and J. Vuillemin, Completeness results for the equivalence of recursive schemes, *J. Comput. System Sci.* **12** (1976) 179–197.

[28] G. Cousineau, La programmation en EXEL, *Rev. Tech. Thomson-CSF* **10** (1978) 209–234 and **11** (1979) 13–35.

[29] G. Cousineau, An algebraic definition for control structures, *Theoret. Comput. Sci.* **12** (1980) 175–192.

[30] W. Damm, The IO- and OI-hierarchies, *Theoret. Comput. Sci.* **20** (1982) 95–207.

[31] W. Damm, E. Fehr and K. Indermark, Higher type recursion and self-application as control structures, in: E. Neuhold, Ed., *Formal Descriptions of Programming Concepts* (North-Holland, Amsterdam, 1978) 461–487.

[32] C. Elgot, Monadic computation and iterative algebraic theories, in: H.E. Rose, Ed., *Logic Colloquium* **73** (North-Holland, Amsterdam, 1975) 175–230.

[33] C. Elgot, Structured programming with and without GOTO statements, *IEEE Trans. Software Engrg.* **2** (1976) 41–54.

[34] C. Elgot, S. Bloom and R. Tindell, The algebraic structure of rooted trees, *J. Comput. System Sci.* **16** (1978) 362–399.

[35] J. Engelfriet and E. Schmidt, IO and OI, *J. Comput. System Sci.* **15** (1977) 328–361 and **16** (1978) 67–99.

[36] P. Enjalbert, Systèmes de déductions pour les arbres et les schémas de programmes, *RAIRO Inform. Théor.* **14** (1980) 247–278 and **15** (1981) 3–21.

[37] J. Gallier, DPDA's in 'atomic normal form' and applications to the equivalence problems, *Theoret. Comput. Sci.* **14** (1981) 155–186.

[38] J. Gallier, Recursion closed algebraic theories, *J. Comput. System Sci.*, to appear.

[39] J. Gallier, *N*-rational algebras, I: Basic Properties and free algebras, II: Varieties and the logic of inequalities, Submitted for publication.

[40] S. Ginali, Regular trees and the free iterative theory, *J. Comput. System Sci.* **18** (1979) 228–242.

[41] J. Goguen, J. Thatcher, E. Wagner and J. Wright, Initial algebra semantics and continuous algebras, *J. ACM* **24** (1977) 68–95.

[42] S. Gorn, Explicit definitions and linguistic dominoes, in: J. Hart and S. Takasu, Eds., *Systems and Computer Science* (University of Toronto Press, 1967) 77–105.

[43] I. Guessarian, Program transformations and algebraic semantics, *Theoret. Comput. Sci.* **9** (1979) 39–65.

[44] I. Guessarian, *Algebraic Semantics*, Lecture Notes in Computer Science **99** (Springer, Berlin, 1981).

[45] M. Harrison, *Introduction to Formal Language Theory* (Addison-Wesley, Reading, MA, 1978).

[46] M. Harrison, I. Havel and A. Yehudai, On equivalence of grammars through transformation trees, *Theoret. Comput. Sci.* **9** (1979) 173–205.

[47] S. Heilbrunner, An algorithm for the solution of fixed-point equations for infinite words, *RAIRO Inform. Theor.* **13** (1979) 131–141.

[48] G. Huet, Résolution d'équations dans les langages d'ordre $1, 2, \ldots \omega$, Doctoral dissertation, University Paris-7 (1976).

[49] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems, *J. ACM* **27** (1980) 797–821.

[50] J. Leszczylowski, A theorem on resolving equations in the space of languages, *Bull. Acad. Polon. Sci., Ser. Sci. Math. Astronom. Phys.* **19** (1979) 967–970.

[51] G. Markowsky and B. Rosen, Bases for chain-complete posets, *IBM J. Res. Develop.* **20** (1976) 138–147.

[52] J. Mycielski and W. Taylor, A compactification of the algebra of terms, *Algebra Universalis* **6** (1976) 159–163.

[53] M. Nivat, On the interpretation of recursive polyadic program schemes, Symposia Mathematica **15** (Academic Press, New York, 1975) 255–281.

[54] M. Nivat, Mots infinis engendrés par une grammaire algébrique, *RAIRO Inform. Théor.* **11** (1977) 311–327.

[55] M. Nivat, Private communication.

[56] L. Nolin and G. Ruggin, A formalization of EXEL, *Proc. ACM Symposium on Principles of Programming Languages*, Boston (1973).

[57] M. O'Donnell, *Computing in Systems Described by Equations*, Lecture Notes in Computer Science **58** (Springer, Berlin, 1977).

[58] M. Oyamaguchi and N. Honda, The decidability of the equivalence for deterministic stateless pushdown automata, *Information and Control* **38** (1978) 367–376.

[59] M. Oyamaguchi, N. Honda and Y. Inagaki, The equivalence problem for real-time strict deterministic languages, *Information and Control* **45** (1980) 90–115.

[60] M. Paterson and M. Wegman, Linear unification, *J. Comput. System Sci.* **16** (1978) 158–167.

[61] J. Robinson, A machine-oriented logic based on the resolution principle, *J. ACM* **12** (1965) 23–41.

[62] B. Rosen, Tree-manipulating systems and Church–Rosser theorems, *J. ACM* **20** (1973) 160–187.

[63] B. Rosen, Program equivalence and context-free grammars, *J. Comput. System Sci.* **11** (1975) 358–374.

[64] M. Schützenberger, On context-free languages and push-down automata, *Information and Control* **6** (1963) 246–264.

[65] J. Tiuryn, Fixed points and algebras with infinitely long expressions, *Fund. Inform.* **II** (1978) 107–128 and **II** (1979) 317–335.

[66] J. Tiuryn, On a connection between regular algebras and rational algebraic theories, *Proc. 2nd Workshop on Categorical and Algebraic Methods in Computer Science and System Theory*, Dortmund, West Germany (1978).

[67] J. Tiuryn, Unique fixed points vs. least fixed points, Report 49, RWTH Aachen, West Germany (1978).
[68] L. Valiant, The equivalence problem for deterministic finite-turn push-down automata, *Information and Control* **25** (1974) 123–133.
[69] J. Wright, J. Thatcher, E. Wagner and J. Goguen, Rational algebraic theories and fixed-point solutions, *Proc. 17th Symposium on Foundations of Computer Science*, Houston, TX (1976) 147–158.
[70] J. Wright, E. Wagner and J. Thatcher, A uniform approach to inductive posets and inductive closure, *Theoret. Comput. Sci.* **7** (1978) 57–77.

Theoretical Computer Science 30 (1984) 205–239
North-Holland

THE SOLUTIONS OF TWO STAR-HEIGHT PROBLEMS FOR REGULAR TREES

J.P. BRAQUELAIRE and B. COURCELLE
Department of Mathematics and Computer Science, Bordeaux-I University, 33405 Talence, France

Introduction

Regular trees, i.e., trees which are either finite or infinite with only finitely many distinct subtrees, play an important role in the theory of program schemes. They have been investigated by Cousineau [9], Jacob [16], Elgot et al. [13] and Courcelle [8].

Since they form the free iterative theory (generated by some ranked alphabet F), they are denoted by certain *iterative theory expressions* (see [2, 13, 14]). These iterative theory expressions include the *rational expressions* independently defined by Cousineau [9]. The relation between these two classes of expressions has been shown by Courcelle [8].

All these expressions use an iteration operator (denoted † or $*$) very close to Kleene's $*$ for languages. They raise a *star-height problem*, i.e., the problem of constructing a rational expression of minimal star-height which defines a given regular tree.

This problem is trivial for iterative theory expressions which use vector iteration since every regular tree can be defined by such an expression with one iteration if the tree is infinite and no iteration if it is finite [12]. It is not if iterative theory expressions are restricted so as to use only *scalar iteration*. We solve it and we show that the minimal star-height is exactly the rank of the minimal graph of the tree (the rank of a directed graph has been introduced by Eggan [10] for the study of rational expressions defining languages and further investigated by McNaughton [17, 18] and Cohen and Brzozowski [3–6]).

These expressions use an operation called *composition*, a typical case of which is $e._n(e_1, \ldots, e_n)$, which denotes the tree obtained by the substitution of $\mathbf{Val}(e_1), \ldots, \mathbf{Val}(e_n)$ at certain leaves of $\mathbf{Val}(e)$ (we denote by $\mathbf{Val}(e)$ the tree defined by the expression e).

The major contribution of Cousineau was to show that the operation of composition is dispensable and that the resulting expressions still generate all regular trees (see [8] for a simple proof). These restricted expressions raise another star-height problem for which we also give the solution. The minimal star-height in this sense is also obtained from the consideration of the minimal graph of the tree.

For technical reasons, we shall work neither with iterative theory expressions [2, 8, 13] nor with rational expressions [8, 9] but with slightly different expressions (still called *rational*) which use the following constructions:

$*_v(e)$: iterate **Val**(e) with respect to the variable v.

$e_{v_1,\dots,v_k}(e_1, \dots, e_k)$: substitute **Val**$(e_1), \dots,$ **Val**(e_k) for $v_1, \dots, v_k$ in **Val**(e).

Our results will be obtained for these rational expressions but they transfer easily to the above mentioned expressions.

The proofs of our two results follow the same pattern that can be sketched as follows.

A regular tree is manipulated by means of a finite pointed graph of which it is the infinite unlooping. These graphs can be 'structured', in different ways, but each 'structuring' is characterized by an integer, its 'depth'.

For each structuring of 'depth' n, one can construct a rational expession of star-height n. Hence, a certain rational expression can be associated with a 'structuring' of minimal 'depth' of the minimal pointed graph of the given tree.

It turns out that this rational expression is the right one, i.e., is of minimal star-height among all those defining the given tree.

In order to prove this, we first define some syntactical manipulations performing some simplifications of rational expressions. They transform a rational expression into an equivalent one in *normal form*.

From the syntactical structuring of a minimal rational expression in normal form, one can construct a 'structuring' of the minimal graph of the tree whose 'depth' is not less than the star-height of the expression. And from the first construction one gets an equality as required.

References

[1] H. Barendregt, *The Lambda-Calculus*, Studies in Logic **103** (North-Holland, Amsterdam, 1981).

[2] S. Bloom and C. Elgot, The existence and construction of free iterative theories *J. Comput. System Sci.* **12** (1976) 305–318.

[3] R. Cohen, Star-height of certain families of regular events, *J. Comput. System Sci.* **4** (1970) 281–297.

[4] R. Cohen, Techniques for establishing star-height of regular sets, *Math. Systems Theory* **5** (1971) 97–114.

[5] R. Cohen, Rank non-increasing transformations on transition graphs, *Inform. and Control* **20** (1972) 93–113.

[6] R. Cohen and J. Brzozowski, General properties of star-height of regular events, *J. Comput. System Sci.* **4** (1970) 260–280.

[7] B. Courcelle, A representation of trees by languages Part I, *Theoret. Comput. Sci.* **6** (1978) 255–279; Part II, *Theoret. Comput. Sci.* **7** (1978) 25–55.

[8] B. Courcelle, Fundamental properties of infinite trees, *Theoret. Comput. Sci.* **25** (1983) 95–169.

[9] G. Cousineau, An algebraic definition for control structures, *Theoret. Comput. Sci.* **12** (1980) 175–192.

[10] L. Eggan, Transition graphs and the star-height of regular events, *Michigan Math. J.* **10** (1963) 385–397.

[11] S. Eilenberg, *Automata, Languages and Machines, Vol. A* (Academic Press, New York, 1974).

[12] C. Elgot, Monadic computations and iterative algebraic theories, in: H.E. Rose, ed., *Logic Colloquium* '73 (North-Holland, Amsterdam, 1975) pp. 175–230.

[13] C. Elgot, S. Bloom and R. Tindell, The algebraic structure of rooted trees, *J. Comput. System Sci.* **16** (1978) 362–399.

[14] S. Ginali, Regular trees and the free iterative theory, *J. Comput. System Sci.* **18** (1979) 228–242.

[15] G. Huet, Confluent reductions: abstract properties and applications to term rewriting systems, *J. Assoc. Comput. Mach.* **27** (1980) 797–821.

[16] G. Jacob, Calcul du rang des arbres infinis réguliers, *Proc. CAAP'81*, Lecture Notes Comput. Sci. **112** (Springer, Berlin, 1981) pp. 238–254.

[17] R. McNaughton, The loop complexity of pure-group events, *Inform. Control* **11** (1967) 167–176.

[18] R. McNaughton, The loop complexity of regular events, *Inform. Sci.* **1** (1969) 305–328.

B. COURCELLE

U.E.R. de Mathematiques et Informatique

Université de Bordeaux I - F-33405 TALENCE