Journal of Automata, Languages and Combinatorics **7** (2002) 1, 11–70

BOTTOM-UP AND TOP-DOWN TREE SERIES TRANSFORMATIONS

JOOST ENGELFRIET
LIACS, Leiden University
PO Box 9512, NL-2300 RA Leiden, The Netherlands
e-mail: engelfri@liacs.nl

ZOLTÁN FÜLÖP [1]
Department of Computer Science, University of Szeged
Árpád tér 2., H-6720 Szeged, Hungary
e-mail: fulop@inf.u-szeged.hu

and

HEIKO VOGLER
Faculty of Computer Science, Technical University of Dresden
D-01062 Dresden, Germany
e-mail: vogler@inf.tu-dresden.de

ABSTRACT

We generalize bottom-up tree transducers and top-down tree transducers to the concept of *bottom-up tree series transducer* and *top-down tree series transducer*, respectively, by allowing formal tree series as output rather than trees, where a formal tree series is a mapping from output trees to some semiring. We associate two semantics with a tree series transducer: a mapping which transforms trees into tree series (for short: tree to tree series transformation or t-ts transformation), and a mapping which transforms tree series into tree series (for short: tree series transformation or ts-ts transformation).

We show that the standard case of tree transducers is reobtained by choosing the boolean semiring under the t-ts semantics. Moreover, we show that certain fundamental constructions and results concerning bottom-up and top-down tree transducers can be generalized for the corresponding tree series transducers. Among others, we prove that polynomial bottom-up t-ts transformations can be characterized by the composition of finite state relabeling t-ts transformations and boolean homomorphism t-ts transformations. Moreover, we prove that every deterministic top-down t-ts transformation can be characterized by the composition of a boolean homomorphism t-ts transformation and a deterministic linear top-down t-ts transformation. We prove that deterministic top-down t-ts transformations are closed under right composition with nondeleting and linear deterministic top-down t-ts transformations and are closed under left composition with boolean and total deterministic top-down t-ts transformations. Finally we

[1] Research of this author was supported by the Hungarian Scientific Foundation (OTKA) under Grant T 030084

show that nondeleting linear bottom-up and nondeleting linear top-down tree series transducers generate the same t-ts transformation class.

Keywords: Tree transducers, weighted automata, semirings, formal power series

1. Introduction

The investigation of this paper was inspired by [28] in which a first attempt was made to integrate formal power series over trees into a restricted form of top-down tree transducer. The resulting tree series transducers transform an input tree into a formal tree series which is a mapping from the set of output trees into some semiring. By this means it is possible to measure the computation of output trees. Here we introduce the concepts of bottom-up tree series transducer and of (unrestricted) top-down tree series transducer. Before we discuss our investigation, we will now briefly review the origins of tree series transducers which are a) tree transducers and b) automata with multiplicity (or cost functions).

Tree transducers have been introduced in [30, 31, 36, 37]. They can be viewed as generalizations of generalized sequential machines [3] to trees where the trees are either read and processed from their leaves towards their roots (bottom-up tree transducer) or from their roots towards their leaves (top-down tree transducer). A rule (or: transition) of a bottom-up tree transducer looks like

$$\sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(t)$$

and a rule of a top-down tree transducer has the form

$$q(\sigma(x_1, \ldots, x_k)) \to \zeta$$

where $q, q_1, \ldots, q_k$ are states of the tree transducers, σ is an input symbol of rank k, $x_1, \ldots, x_k$ are variables ranging over trees (more precisely, over output trees in the bottom-up case, and over input trees in the top-down case), $t \in T_\Delta(X_k)$ is an output tree which may contain variables from the set $X_k = \{x_1, \ldots, x_k\}$, and ζ is an output tree which may contain constructs of the form $p(x_i)$ at its leaves where p is a state and $1 \leq i \leq k$. In the usual way, the rules of a tree transducer constitute a binary term rewriting relation (or: derivation relation) $\Rightarrow_M$ by means of which the tree transformation $\tau_M \subseteq T_\Sigma \times T_\Delta$ computed by M can be defined (where T_Σ and T_Δ are the sets of input trees and output trees, respectively).

Since the seventies, tree transducers have been studied intensively. Some of the first papers dealt with (de-)composition and hierarchy results [1, 11, 12, 13]. In [20] a method of deciding the equivalence of the compositions of classes of tree transformations is overviewed. Survey articles and books are [22, 23, 7, 21]. Recently, a characterization of tree transformation classes in terms of monadic second order logic has been proved [2, 10].

Now let us briefly review the second origin of tree series transducers: automata with multiplicity. Let $M = (Q, \Sigma, \mu, q_0, Q_d)$ be a usual finite state string automaton where Q is the set of states, Σ the set of input symbols, $\mu : Q \times \Sigma \to \mathcal{P}(Q)$ the transition function, $q_0 \in Q$ the initial state, and $Q_d \subseteq Q$ the set of final states. In the usual