Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 347 (2005) 276–287

Theoretical Computer Science

www.elsevier.com/locate/tcs

Linear deterministic multi bottom-up tree transducers☆

Zoltán Fülöp[a,*], Armin Kühnemann[b], Heiko Vogler[b]

[a] *Department of Computer Science, University of Szeged, Árpád tér 2., H-6720 Szeged, Hungary*
[b] *Faculty of Computer Science, Technical University of Dresden, D-01062 Dresden, Germany*

Received 10 September 2004; received in revised form 14 January 2005; accepted 22 July 2005

Communicated by M. Ito

Abstract

In general, top-down and bottom-up tree transducers lead to incomparable classes of tree transformations, both for the nondeterministic and the deterministic case. If deterministic top-down tree transducers are extended by the capability to recognize regular tree properties and deterministic bottom-up tree transducers are generalized by allowing states with arbitrary finite rank, then the two devices, now called deterministic top-down tree transducers with regular look-ahead and deterministic multi bottom-up tree transducers, respectively, become equivalent [Z. Fülöp, A. Kühnemann, H. Vogler, A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead, Inform. Process. Lett. 91 (2004) 57–67].

In this paper we focus on the class *ld-MBOT* of tree transformations which are computed by linear deterministic multi bottom-up tree transducers. We investigate the relationship among *ld-MBOT* and the classes of tree transformations computed by (restricted) deterministic bottom-up tree transducers and by (restricted) deterministic top-down tree transducers with regular look-ahead. In fact, we show the inclusion diagram of nine such classes.

Keywords: Tree automata; Tree transducers

☆ This research was supported by the Hungarian Scientific Foundation (OTKA) under Grant T 046686, the Exchange Programme of the DAAD and MÖB (project "Weighted Tree Automata", Grant No. 36) and the Herbert-Quandt-Foundation.

* Corresponding author. Tel.: +36 62 544 302; fax: +36 62 546 397.
E-mail addresses: fulop@inf.u-szeged.hu (Z. Fülöp), kuehne@tcs.inf.tu-dresden.de (A. Kühnemann), vogler@tcs.inf.tu-dresden.de (H. Vogler).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2005.07.025

1. Introduction

In the theory of tree automata and tree transducers bottom-up (or frontier-to-root) devices and top-down (or root-to-frontier) devices are considered: in the former, the processing starts at the leaves of an input tree s and ends up at the root of s, and in the latter, the processing starts at the root of s and proceeds towards the leaves of s (cf., e.g., [2,11,12,15,16]). A number of investigations have been done in order to compare the accepting or transformational power of the two alternative processing modes of a device. For instance, the class of tree languages accepted by deterministic top-down tree automata is a proper subclass of the class of tree languages accepted by deterministic bottom-up tree automata (cf., e.g. [11, Example 2.11 in Chapter II]); or the classes of tree transformations computed by bottom-up tree transducers and by top-down tree transducers are incomparable (cf. [2, Theorem 2.3]); or: the classes of tree transformations computed by bottom-up tree-to-graph transducers and by top-down tree-to-graph transducers are equal (cf. [8, Theorem 7.1]).

In the case that the two classes are not equal, features have been added to the devices in order to remedy the deficiencies of the respective models. For instance, linear top-down tree transducers have a strictly smaller transformational power than linear bottom-up tree transducers (cf. [2, Theorem 2.8]); however, if linear top-down tree transducers are equipped with regular look-ahead, then they are as powerful as linear bottom-up tree transducers (cf. [3, Theorem 2.8]). Another example is the following: the classes of tree transformations computed by deterministic bottom-up tree transducers and by deterministic top-down tree transducers are incomparable (cf. [3, Section 3]); however, if deterministic bottom-up tree transducers are generalized by allowing the states to have an arbitrary finite rank and the deterministic top-down tree transducers are equipped with regular look-ahead, then the classes of tree transformations become equal (cf. [10, Theorem 4.4]). Such generalized bottom-up tree transducers are called *multi bottom-up tree transducers*.

In this paper we deal with deterministic multi bottom-up tree transducers. Roughly speaking, a deterministic multi bottom-up tree transducer M is a term rewriting system which is based on the three ranked alphabets of states, input symbols, and output symbols. There are rules processing input symbols. These rules have the form

$$\sigma(q_1(x_{1,1}, \ldots, x_{1,n_1}), \ldots, q_k(x_{k,1}, \ldots, x_{k,n_k})) \to q_0(t_1, \ldots, t_n),$$

where $k \geqslant 0$, σ is a k-ary input symbol, $q_0, q_1, \ldots, q_k$ are states of rank $n, n_1, \ldots, n_k$, respectively, and the $t_1, \ldots, t_n$ are trees over the ranked alphabet of output symbols and the variables $x_{1,1}, \ldots, x_{1,n_1}, \ldots, x_{k,1}, \ldots, x_{k,n_k}$. Moreover, in order to guarantee determinism, we require that there are no different rules with the same left-hand side. Now, the transformation $\tau_M(s)$ of an input tree s is computed as follows: first, a special unary symbol *root* is put on top of s; then, using the above rules, the transducer computes a tree of the form $root(q(u_1, \ldots, u_m))$ where q is an m-ary state and $u_1, \ldots, u_m$ are output trees; finally, a rule of the form

$$root(q(x_1, \ldots, x_m)) \to q_f(t)$$

is applied, where q_f is a special unary symbol called final state and t is a tree over the ranked alphabet of output symbols and the variables $x_1, \ldots, x_m$. Again, there are no different