Erratum to: "Top-down Tree Transducers with Regular Look-ahead".

Engelfriet, J.

Theory Comput Syst (2016) 58:377–379
DOI 10.1007/s00224-015-9647-z

ERRATUM

Erratum to: "Top-down Tree Transducers with Regular Look-ahead"

Joost Engelfriet[1]

Published online: 29 August 2015

In Theorem 3.2 of my paper [1] it is shown that *DB-FST* $\subset$ DT^R*-FST*, where *DB-FST* is the class of deterministic bottom-up finite state tree transformations and DT^R*-FST* is the class of deterministic top-down finite state tree transformations with regular look-ahead (and $\subset$ denotes proper inclusion). One may then ask for a characterization of *DB-FST* in terms of dt^r*-fst*. In Theorem 3.2 of [1] it is wrongly stated that *DB-FST* = ODT^R*-FST*, where ODT^R*-FST* is the class of one-state dt^r*-fst*. The correct statement is that

$$DB\text{-}FST = FTA \circ ODT^R\text{-}FST,$$

where *FTA* is the class of tree transformations that are the identity on a recognizable (= regular) tree language. In other words, a tree transformation is a *db-fst* if and only if it is the restriction of a one-state dt^r*-fst* to a recognizable tree language. Note that this implies that a total function from T_Σ to T_Δ is in *DB-FST* if and only if it is in ODT^R*-FST* (as mentioned in Section 2 of [3]).

Why the proof is wrong. The proof of the inclusion *DB-FST* $\subseteq$ ODT^R*-FST* of Theorem 3.2 is based on the fact that the construction in the proof of Lemma 2.10(3) preserves the number of states. But that is not true because in the latter proof it should be assumed that the initial state q_d of T does not occur in the right-hand side of any rule of T. This can indeed be assumed without loss of generality, but possibly increases the number of states by one.

The online version of the original article can be found at http://dx.doi.org/10.1007/BF01683280.

✉ Joost Engelfriet
j.engelfriet@liacs.leidenuni.nl

1 Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

Why the result is wrong. Since *FTA* $\subset$ *DB-FST*, it suffices to show that *FTA* is not included in *ODT^R-FST*. Suppose that $T \in$ *ODT^R-FST* is the identity on the recognizable tree language $\{\sigma(t) \mid t \in T_\Sigma\}$, where $\Sigma = \{\tau, a\}$, σ and τ have rank 1, and a has rank 0. Let T have the unique state q, and let $t_0 \in T_\Sigma$ be a tree of size larger than the size of the rules of T. Consider the computation of T with input (and output) $\sigma(t_0)$. The first rule applied in this computation must be of the form $\langle q(\sigma(x)) \to s, D\rangle$ such that $t_0 \in D(x)$ and $q(x)$ occurs in s. Thus, there must be a successful computation of T starting with $q(t_0)$. That contradicts the fact that t_0 is not in the domain of T. Hence *ODT^R-FST* $\subset$ *DB-FST*, where the inclusion is proper.

Why the new result holds. From the inclusion *ODT^R-FST* $\subset$ *DB-FST* it follows that *FTA* $\circ$ *ODT^R-FST* $\subseteq$ *FTA* $\circ$ *DB-FST* $\subseteq$ *DB-FST* $\circ$ *DB-FST* $\subseteq$ *DB-FST* where the last inclusion is Theorem 4.6(2) of [2].

It remains to prove that *DB-FST* $\subseteq$ *FTA* $\circ$ *ODT^R-FST*. We will say that a *db-fst* $\langle\Sigma, \Delta, Q, Q_d, R\rangle$ is *full* if $Q_d = Q$, i.e., every state is final. We will denote the class of all full *db-fst* by *FDB-FST*, and the class of all full deterministic bottom-up finite state relabelings by *FDBQREL*. Note that *FDB-FST* is the class of total functions from T_Σ to T_Δ in *DB-FST*.

Since the domain of a *db-fst* is recognizable (by Corollary 3.12 of [2]), it should be clear that *DB-FST* $\subseteq$ *FTA* $\circ$ *FDB-FST*. Consequently, it now suffices to prove that *FDB-FST* $\subseteq$ *ODT^R-FST*. In fact, the wrong proof of the inclusion *DB-FST* $\subseteq$ *ODT^R-FST* in the proof of Theorem 3.2 is valid for full *db-fst*. It follows from the proof of Theorem 3.15(3) of [2] that *FDB-FST* $\subseteq$ *FDBQREL* $\circ$ *HOM*. Since the identity is in *ODT^R-FST*, this is included in *ODT^R-FST* $\circ$ *FDBQREL* $\circ$ *HOM*. Now the (wrong) construction in the proof of Lemma 2.10(3), discussed above, is actually correct for *FDBQREL*. Since that construction, and the one in the proof of Lemma 2.9, preserves the number of states, we obtain that *ODT^R-FST* $\circ$ *FDBQREL* $\circ$ *HOM* $\subseteq$ *ODT^R-FST*. This shows that *FDB-FST* $\subseteq$ *ODT^R-FST* and hence *DB-FST* $=$ *FTA* $\circ$ *ODT^R-FST*.

Remark. Let *O'DT^R-FST* be the class of *dt^r-fst* with two states, such that the initial state does not occur in the right-hand side of any rule. It is not difficult to show that *DB-FST* $\subseteq$ *O'DT^R-FST*. In fact, the wrong proof of the inclusion *DB-FST* $\subseteq$ *ODT^R-FST* in the proof of Theorem 3.2 is valid for *O'DT^R-FST*, because the constructions in the proofs of Lemmas 2.10(3) and 2.9 preserve *O'DT^R-FST*. It is also easy to see that the inclusion is proper, i.e., *DB-FST* $\subset$ *O'DT^R-FST*, as witnessed by the tree transformation $\{(t, \sigma(t)) \mid t \in T_\Sigma\}$.

Three other small corrections. Six lines above Theorem 2.6 of [1] it is stated that *ZT-FST* $\subset$ *ZT-FST*; that should of course be *ZT-FST* $\subset$ *ZT^R-FST*. On the 4th line of the proof of Lemma 2.10 of [1], the equation $K = T\,L$ should be $K = T \circ L$. After Theorem 2.11 of [1] it is stated that the inclusion signs in Theorem 2.6 may be replaced by equality signs; to see this, one should note that *DBQREL* $\subseteq$ *LDT^R-FST* (by the proof of Lemma 2.10(3)).

Acknowledgments I am grateful to Heiko Vogler for pointing out my mistake (a long time ago). I thank one of the reviewers of this corrigendum for helpful comments.

References

1. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Mathematical Systems Theory **10**, 289–303 (1977)
2. Engelfriet, J.: Bottom-up and top-down tree transformations – a comparison. Mathematical Systems Theory **9**, 198–231 (1975)
3. Engelfriet, J.: Tree transducers and syntax-directed semantics. Proceedings of the 7th Colloquium on Trees in Algebra and Programming (CAAP), Lille (1982)