Chapter 5

Tree Transducers

5.1 Deterministic Bottom-up Finite-state Tree Transducers

5.1.1 Orientation

This section is about deterministic bottom-up finite-state tree transducers. The term *finite-state* means that the amount of memory needed in the course of computation is independent of the size of the input. The term *deterministic* means there is single course of action the machine follows to compute its output. The term *transducer* means this machine solves *transformation problem*: given an input object x, what object y is x transformed into? The term *tree* means we are considering the transformation problem from trees to trees. The term *bottom-up* means that for each node a in a tree, the computation transforms the children of a node before transforming the node. This contrasts with *top-down* transducers which transform the children after transforming their parent. Visually, these terms make sense provided the root of the tree is at the top and branches of the tree move downward.

A definitive reference for finite-state automata for trees is freely available online. It is "Tree Automata Techniques and Applications" (TATA) (Comon *et al.*, 2007). The presentation here differs from the one there, as mentionepd below.

5.1.2 Definitions

Recall the definition of trees with a finite alphabet Σ and the symbols [ ] not in Σ. All such trees belonged to the treeset Σ^T. In addition to this, we will need to define a new kind of tree which has *variables* in the leaves. I will call these trees *Variably-Leafed*. We assume a countable set of variables X containing variables $x_1, x_2, \ldots$.

Definition 26 (Variably-Leafed Trees)**.**

Base Cases (Σ): *For each $a \in \Sigma$, $a[\ ]$ is a tree.*
Base Cases (X): *For each $x \in X$, $x[\ ]$ is a tree.*

Inductive Case: *If $a \in \Sigma$ and $t_1 t_2 \ldots t_n$ is a string of trees of length n then $a[t_1 t_2 \cdot \ldots t_n]$ is a tree.*

Let $\Sigma^T[X]$ denote the set of all variably-leafed trees of finite size using Σ and X.

Note that $\Sigma^T \subsetneq \Sigma^T[X]$. In the tree transducers we write below, the variably-leafed trees will play a role in the omega function as well as the the intermediate stages of the transformation.

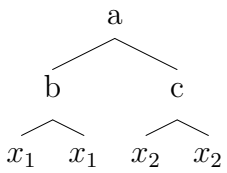
With this definition in place, we can define our first tree transducer.

Definition 27 (DBFTA)**.** *A* Deterministic Bottom-up Finite-state Acceptor (DBFTT) *is a tuple (Q, Σ_r, F, δ) where*

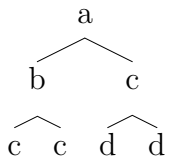
- *Q is a finite set of states;*
- *Σ is a finite alphabet;*
- *$F \subseteq Q$ is a set of* accepting *(*final*) states; and*
- *$\delta : Q^* \times \Sigma \to Q$ is* the transition function. The pre-image of δ must be finite. *This means we can write it down—for example, as a list.*
- *Ω is a function with domain $Q^* \times \Sigma$ and co-domain $\Sigma^T[X]$.* Its pre-image must also be finite.

Generally, the pre-images of δ and Ω should coincide.

Example 11. Let M be a DBFTT and suppose $\Omega(q_1, q_2, a) = a\big[\ b[x_1\ x_1]\ \ c[x_2\ x_2]\ \big]$. So this is the variably-leafed tree shown below.

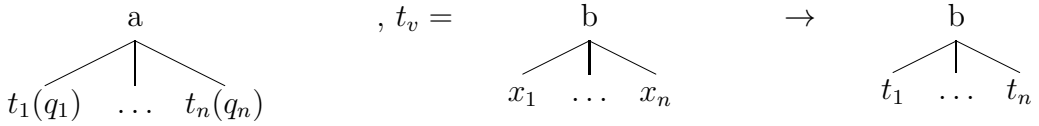


Further suppose $\Omega(\lambda, c) = c[\ ]$ and $\Omega(\lambda, d) = d[\ ]$. The idea is that M will transform the tree shown below at left into the tree shown below at right.

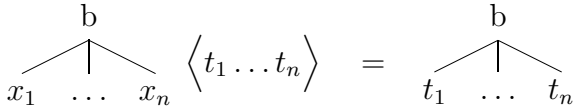


Intuitively, this is because each variable x_i in the variably leafed tree will be replaced by the ith child of the root node of the tree under consideration.

Here is the general schema. Given $\Omega(q_1 q_2 \ldots q_n, a) = t_v$ and a tree $a[t_1 t_2 \ldots t_n]$ with states $q_1 q_2 \ldots q_n$, respectively, then the output tree will be the one obtained by replacing each x_i with t_i in t_v. A schematic of this is shown below.



We now formalize the above ideas. As before, we extend the transition function δ to $\delta^* : \Sigma^T \to Q$. If $t_1 \dots t_n$ is a list of trees and $t_v \in \Sigma^T[X]$ is a variably leafed tree with variables $x_1, \dots x_n$ then let $t_v\langle t_1 \dots t_n \rangle = t \in \Sigma^T$ obtained by replacing each variable x_i in t_v with t_i. Here is another visualization with this notation.



We also define a new function "process" $\pi : \Sigma^T \to Q \times \Sigma^T$ which will process the tree and produce its output. It is defined as follows.

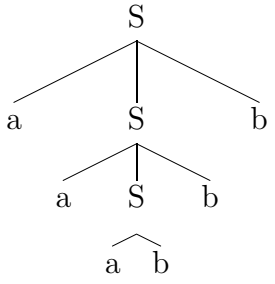
$$\begin{aligned} \pi(a[\,]) &= \big(\delta(\lambda, a), \Omega(\lambda, a)\big) \\ \pi(a[t_1 \cdots t_n]) &= (q, t) \\ &\quad \text{where } q = \delta\big(q_1 \cdots q_n, a\big) \\ &\quad \text{and } t = \Omega\big(q_1 \cdots q_n, a\big)\langle s_1 \cdots s_n \rangle \\ &\quad \text{and } (q_1, s_1) \cdots (q_n, s_n) = \pi(t_1) \cdots \pi(t_n) \end{aligned} \tag{5.1}$$

Definition 28 (Tree-to-tree function of a DBFTT)**.** *The function defined by the transducer M is $\big\{(t, s) \mid t, s \in \Sigma^T, \pi(t) = (q, s), q \in F\big\}$. If (t, s) belongs to this set, we say M* transduces *t to s and write $M(t) = s$.*

Example 12. Consider the transducer M constructed as follows.

- $Q = \{q_a, q_b, q_S\}$
- $\Sigma = \{a, b, S\}$
- $F = \{q_S\}$
- $\delta(\lambda, a) = q_a$
- $\delta(\lambda, b) = q_b$
- $\delta(q_a q_b, S) = q_S$
- $\delta(q_a q_S q_b, S) = q_S$
- $\Omega(\lambda, a) = a[\,]$
- $\Omega(\lambda, b) = b[\,]$
- $\Omega(q_a q_b, S) = S[x_2 x_1]$
- $\Omega(q_a q_S q_b, S) = S[x_3 x_2 x_1]$

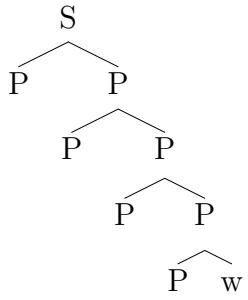
Let us work out what M transforms the tree below into.

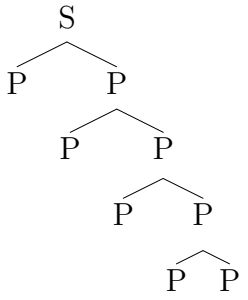


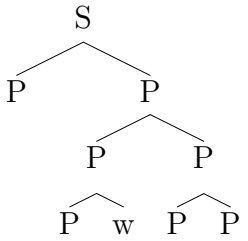
Example 13. Consider the transducer M constructed as follows. We let $Q = \{q_w, q_p, q_S\}$, $\Sigma = \{w, P, S\}$, and $F = \{q_S\}$. The transition and ouput functions are given below.

- $\delta(\lambda, P) = q_p$
- $\delta(\lambda, w) = q_w$
- $\delta(q_p q_p, P) = q_p$
- $\delta(q_w q_p, P) = q_w$
- $\delta(q_p q_w, P) = q_w$
- $\delta(q_p q_w, S) = q_S$
- $\delta(q_w q_p, S) = q_S$
- $\delta(q_p q_p, S) = q_S$
- $\Omega(\lambda, P) = P[\ ]$
- $\Omega(\lambda, w) = w[\ ]$
- $\Omega(q_p q_p, P) = P[x_1 x_2]$
- $\Omega(q_w q_p, P) = P[x_1 x_2]$
- $\Omega(q_p q_w, P) = P[x_1 x_2]$
- $\Omega(q_p q_w, S) = S[\ w[\ ]\ S[x_1 x_2]\ ]$
- $\Omega(q_w q_p, S) = S[\ w[\ ]\ S[x_1 x_2]\ ]$
- $\Omega(q_p q_p, S) = S[x_1 x_2]$

Let us work out how M transforms the trees below.







5.2 Deterministic Top-down Finite-state Tree Transducers

5.2.1 Orientation

This section is about deterministic bottom-up finite-state tree transducers. The term *finite-state* means that the amount of memory needed in the course of computation is independent of the size of the input. The term *deterministic* means there is single course of action the machine follows to compute its output. The term *transducer* means this machine solves *transformation problem*: given an input object x, what object y is x transformed into? The term *tree* means we are considering the transformation problem from trees to trees. The term *top-down* means that for each node a in a tree, the computation transforms the node before transforming its children. This contrasts with *bottom-up* transducers which transform the children before transforming their parent. Visually, these terms make sense provided the root of the tree is at the top and branches of the tree move downward.

A definitive reference for finite-state automata for trees is freely available online. It is "Tree Automata Techniques and Applications" (TATA) (Comon *et al.*, 2007). The presentation here differs from the one there, as mentioned below.

5.2.2 Definitions

As before, we use variably leafed trees $\Sigma^T[X]$.

Definition 29 (DTFTT)**.** *A* Deterministic Top-down Finite-state Acceptor (DTFTT) *is a tuple* $(Q, \Sigma_r, q_0, \delta)$ *where*

- Q *is a finite set of states;*
- Σ *is a finite alphabet;*
- $q_0 \in Q$ *is the initial state; and*
- $\delta : Q \times \Sigma \times \mathbb{N} \to Q^*$ *is* the transition function.
- Ω *is a function with domain* $Q \times \Sigma \times \mathbb{N}$ *and co-domain* $\Sigma^T[X]$.

Generally, the pre-images of δ *and* Ω *should coincide.*

We also define a new function "process" $\pi : Q \times \Sigma^T \to \Sigma^T$ which will process the tree and produce its output. It is defined as follows.

$$\begin{aligned} \pi(q, a[\,]) &= \Omega(q, a, 0) \\ \pi(q, a[t_1 \cdots t_n]) &= \Omega(q, a, n)\big\langle \pi(q_1, t_1) \cdots \pi(q_n, t_n) \big\rangle \\ &\quad \text{where } q_1 \cdots q_n = \delta(q, a, n) \end{aligned} \tag{5.2}$$

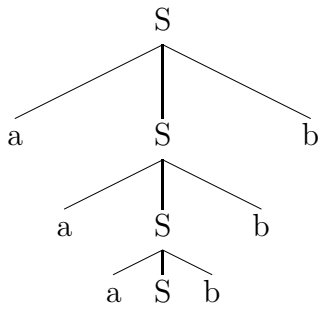
Intuitively, Ω transforms the root node into a variably leafed tree. The variables are replaced with the children of the root node. These children are also trees with states assigned by δ. Then π transforms each tree-child as well.

Definition 30 (Tree-to-tree function of a DTFTT)**.** *The function defined by the transducer* M *is* $\big\{(t, s) \mid t, s \in \Sigma^T, \pi(q_0, t) = s\big\}$*. If* (t, s) *belongs to this set, we say* M transduces t *to* s *and write* $M(t) = s$.

Example 14. Consider the transducer M constructed as follows.

- $Q = \{q, q_S\}$
- $\Sigma = \{a, b, S\}$
- $q_0 = q_S$
- $\delta(q, a, 0) = \lambda$
- $\delta(q, b, 0) = \lambda$
- $\delta(q_S, S, 3) = qq_Sq$
- $\delta(q_S, S, 2) = qq$
- $\Omega(q, a, 0) = a[\,]$
- $\Omega(q, b, 0) = b[\,]$
- $\Omega(q, S, 3) = S[x_3x_2x_1]$
- $\Omega(q, S, 2) = S[x_2x_1]$

Let see how M transforms the tree below.



Exercise 17. Recall the "wh-movement" example from before. Explain why this transformation *cannot* be computed by a deterministic top-down tree transducer.

5.3 Theorems about Deterministic Tree Transducers

Theorem 13 (composition closure)**.** *The class of deterministic bottom-up transductions is closed under composition, but the class of top-down deterministic transductions is not.*

Theorem 14 (Incomparable)**.** *The class of deterministic bottom-up transductions is incomparable with the the class of top-down deterministic transductions.*

This theorem is based on the same kind of examples which separated the left and rigth sequential functions. Let relations $U = (f^n a, f^n a) \mid n \in \mathbb{N} \cup (f^n b, g^n b) | n \in \mathbb{N}$ and $D = (ff^n a, ff^n a) \mid n \in \mathbb{N} \cup (gf^n a, gf^n b) \mid n \in \mathbb{N}$. U is recognized by a DBFTT but not any DTFTT and D is recognized by a DTFTT but not any DBFTT.