Quantifier-free tree transductions

Shiori Ikawa Akane Ohtaka Adam Jardine
Department of Linguistics
Rutgers University
{shiori.ikawa,akane.ohtaka,adam.jardine}@rutgers.edu

1 Introduction

In this work in progress we discuss issues in extending *quantifier-free (QF) logical transductions* from strings to trees. Input-Strictly-Local (ISL) functions, which form an effective class to describe phonological transformations (Chandlee, 2014; Chandlee and Heinz, 2018) and for projecting tiers for long-distance well-formedness conditions (De Santo and Graf, 2019) have been shown to be characterizable with order-preserving QF transductions (Chandlee and Jardine, 2019). We explore how QF transductions can be extended to trees for the purpose of capturing syntactic phenomena. We show QF tree transductions are incomparable to existing tree transducer classes, but do capture some empirically useful transductions. Also, they may be extended with least-fixed point logics to capture a wider range of phenomena, as has been shown for QF logics in strings (Chandlee and Jardine, 2019).

2 Formal definitions

2.1 Logical transductions

Following Courcelle (1994) and Engelfriet and Hoogeboom (2001), we define transductions as logical interpretations. A *signature* is some set of named functions and relations, and a (finite) *model* in that signature is an instantiation of those functions and relations over some (finite) universe of elements. A transduction from models in one signature to models in another can then be described by defining the relations and functions in the output signature using formulas in a logical language of the input signature.

More specifically, for trees labeled with an input alphabet Σ, we define a function to trees over an output alphabet Γ with a series of monadic predicates $\varphi^c_\gamma(x)$—written in the first-order logic of the input trees, without quantifiers—for each $\gamma \in \Gamma$ and $c \in \mathscr{C}$, where $\mathscr{C}$ is a *copy set* that allows us to build $\texttt{card}(\mathscr{C})$ copies for each element in the input tree. The semantics of a transduction is then that an element t in the input tree has a corresponding element labeled t^c in the output tree if and only if $\varphi^c_\gamma(x)$ is true for t.

2.2 Quantifier-free transductions over trees

As a running example for QF tree transductions, we will use the tier-construction function for case assignments. Vu et al. (2019) analyze case assignment as a local well-formedness condition over a tree 'tier', which is itself a tree with irrelevant information removed. The ungrammaticality of the sentence "*He saw she", is captured with a tier constructed by removing all information except D heads carrying Nom or Acc features, C heads, and their immediate parent nodes, as shown in Figure 1: This sentence is bad because the resulting tier contains the local configuration [• he [• she]], where no C head intervenes between the two Nom-featured D-heads as shown in Figure 1.b. Such tier construction functions are non-capturable with simple eraser function (Heinz et al., 2011), as they refer to the input local context in deciding whether to project a certain node. TSL over this tier is more parallel to the Input-local TSL (ITSL) defined over strings in De Santo and Graf (2019), which utilizes the local information in the construction of tiers by constructing tiers with ISL functions, i.e. QF transductions.

There are several considerations required in extending QF logical transductions to trees. First, in order to capture local information with monadic predicates, QF string transductions were defined in Chandlee and Lindell (forthcoming) and Chandlee and Jardine (2019) using functional signatures, where the element in a string are ordered with predecessor and/or successor functions. For our QF tree transductions we assume an input sig-

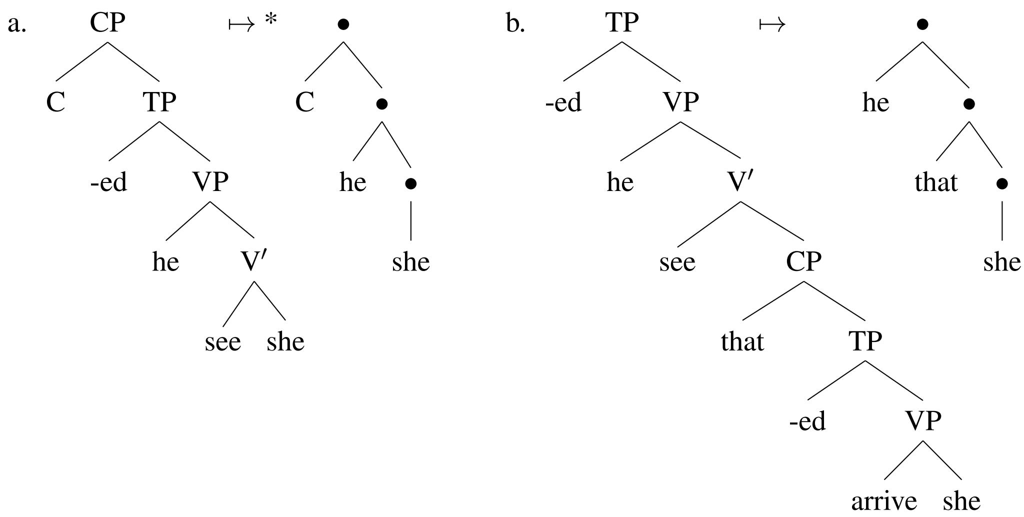


Figure 1: Caption

nature with a parent function μ, where $\mu(x) = y$ when y is the parent node of x, and the predecessor function p, which defines the linear order between sister nodes. Note that we do not use the child relation (i.e. the inverse of μ function), as it is not a function. This means that in (1) we cannot identify the mother nodes of C and D nodes without existentially quantifying the child nodes, so we instead build two copies of C and D nodes themselves.

Second, whereas $\mathscr{C}$ is taken from an initial segment of the natural numbers for string transductions, our copy set $\mathscr{C}$ forms a tree. Members of $\mathscr{C}$ are marked with Gorn address, where the Gorn address of the root will be r. Additionally, exactly one $c \in \mathscr{C}$ will be marked as a 'bottom node' with an additional b label. Every copy tree has to include an r node and b node, as characterized by the well-formedness conditions for a copy tree in (1): When a node exists, the nodes above it including the root node exist (1a) and when a root exists, there is always a bottom node (1b). We will assume that there is at most one root copy r and one bottom copy b. Note that b is a copy to which the lower part of the input tree attaches to, and it does not mean b has to be the lowest node inside $\mathscr{C}$. An example for a copy tree is given in (2a). The case-tier transductions can now be characterized as shown in (2b) and (2c), using the copy tree of the form in (2a).

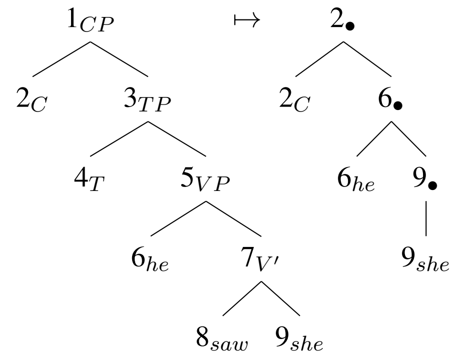
(1) copy well-formedness conditions
For nodes c' and d' s.t. $d' <_\mu c'$,

a. $\varphi^{c'}_D(x) \rightarrow \varphi^{d'}_D(x)$

b. $\varphi^{r}_D(x) \rightarrow \varphi^{b}_D(x)$

(2) a. 

b. $\mathscr{C} := \{rb, 0\}$
$\varphi^{rb}_{\bullet}(x) := C(x) \vee he(x) \vee she(x)$
$\varphi^{0}_{C}(x) := C(x)$
$\varphi^{0}_{he}(x) := he(x)$
$\varphi^{0}_{she}(x) := she(x)$

c. 

2.3 Asymmetric c-command preservation

In a parallel way to how order-preservation in string QF transductions restricts them to regular functions (Filiot, 2015; Chandlee and Jardine, 2019), we will define the structural relationship among the output copies in a way that preserves the structural relation of the input tree: We define the output dominance relation based on the asymmetric c-command in the input, as shown in Table 1a (p. 4): As for the input node x, y $s.t.$ (i) y is dominated by x or (ii) y is asymmetrically c-commanded by x ($higher(x, y)$) and x's parent node and sister node that dominates y, are deleted ($sa\text{-}del(x, y)$), the nodes above bottom node of the copy tree of x dominate all the nodes of the copy tree of y. The latter case serves to keep the asymmetric c-command relation between x and y when the intermediate nodes are deleted. In the copy of the same input node, the domination among nodes is trivially defined. Table 1b shows that the precedence relations in the input trees are preserved among the root nodes of the correspondent copy trees in the output, and the precedence relation among the copies of the same input node is defined trivially.

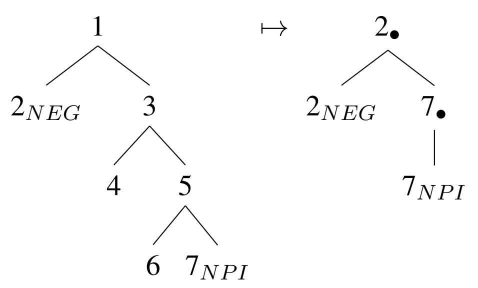
2.4 Comparison with other tree transducers

In general, QF tree transductions as defined here are incomparable to deterministic bottom-up or top-down tree transducers (Comon et al., 2008). Briefly, this is because QF tree transductions get a finite "lookahead" in either direction. However, for this reason, QF tree transductions have some similarities to sensing tree automata (Martens et al., 2008; Graf and De Santo, 2019). Future work will examine this relationship further.

3 Other Examples

3.1 Negative polarity tier construction

The definition of tree transductions discussed above can accommodate the case of negative polarity item (NPI) licensing in English. An NPI such as *anyone* is licensed when it is c-commanded by a downward entailing operator such as negation, as the contrast between "John doesn't like anyone" and "*Anyone doesn't like John" shows. The grammaticality of the sentence "John doesn't like anyone" can be captured with a tier of the form in (3). Crucially, just like the case-tier transduction in (2c), the NPI-tier transduction in (3) is QF-definable using the copy tree in (2a), as shown in (4) (see also Graf and Shafiei 2019).

(3)


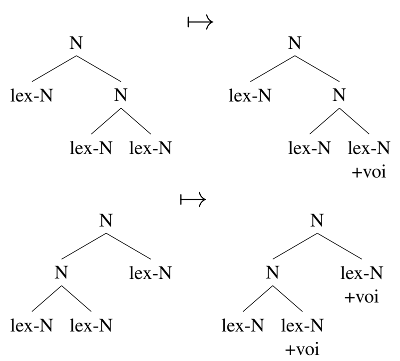
(4)
$$\begin{aligned}\mathscr{C} &:= \{rb, 0\}\\ \varphi^{rb}_{\bullet}(x) &:= NEG(x) \vee NPI(x)\\ \varphi^{0}_{NEG}(x) &:= NEG(x)\\ \varphi^{0}_{NPI}(x) &:= NPI(x)\end{aligned}$$

3.2 Morphological conditioning of *rendaku*

Applicability of tree transductions extends to phonological phenomena as well. Japanese has a phonological operation called *rendaku*, where the first consonant of the second element gets voiced in compounding (e.g. *ao* 'blue'+*sora* 'sky' → *ao-zora* 'blue sky'). There is a structural constraint in this operation in (5) (Otsu, 1980): *sora* does not get voicing when it is in the compound [*ao*-[*sora-mame*]] 'blue broad-bean.' Compounds of the structure [[A B] C] allows their second element to undergo *rendaku* (e.g. [[ao-zora]-yohoo]'forecast of blue sky')

(5) Branching Constraint
Rendaku does not occur on B when the compound has the structure [A [B C]].

The application of a [+voi] feature to a structure can be represented as tree transductions in (6). These transductions are QF-definable as shown in (7): The lex-N node which is not *first* (i.e. the left-most among its sisters) acquires [+voi] feature.

(6) a. b.


(7)
a. $\varphi^{rb}_{\text{N}}(x) := \varphi_{\text{N}}(x)$
b. $\varphi^{rb}_{\text{lex-N}}(x) := \varphi_{\text{lex-N}}(x) \wedge \text{first}(x)$
c. $\varphi^{rb}_{\text{lex-N(+voi)}}(x) :=$
$\varphi_{\text{lex-N}}(x) \wedge \neg\text{first}(x)$
where $\text{first}(x) := p(x) \approx x$

This pattern cannot be captured by (functional) string transductions: Given a string of three lex-N, we cannot decide between the mappings in (8a) and (8b).

(8) lex-N lex-N lex-N
a. ↦ lex-N lex-N(+voi) lex-N(+voi)
b. ↦ lex-N lex-N lex-N(+voi)

For all $c, d \in T_{\mathscr{C}}$ and b and r of $T_{\mathscr{C}}$,

$$\begin{array}{llll}
\text{a.} & x <'^{\,c,d}_{\mu^*} y := & x <_{\mu^*} y \vee (higher(x,y) \wedge \neg\varphi^b_D(\mu(x)) \wedge sa\text{-}del(x,y) & c \leq_{\mu^*} b \\
 & & x \approx y & \text{if } c <_{\mu^*} d \\
 & \text{where} & higher(x,y) := \mu(x) <_{\mu^*} y \wedge \neg\mu(y) <_{\mu^*} x & \\
 & & sa\text{-}del(x,y) := \neg\exists z[sisters(x,z) \wedge \varphi^b_D(z) \wedge z <_{\mu^*} y]) & \\
\text{b.} & x <'^{\,c,d}_{p^*} y := & x <_p y & \text{if } c = d = r \\
 & & x \approx y & \text{if } c <^*_p d
\end{array}$$

Table 1: Formulas for preserving asymmetric c-command

Note that it is not always the case that both of these outputs are grammatical given an input string of three nouns. The examples above illustrate: *ao-zora-yohoo* 'foreceast of blue sky' but *ao-sora-mame* 'blue broad-bean' (cf. **ao-zora-mame*).

4 Future work

Chandlee and Jardine (2019) discuss extending QF logic with least-fixed point operators to capture long-distance processes; a clear next step is to extend this to QF tree transductions. Additionally, for n-branching trees we can study their models with a set of n child functions, instead of the mother function used here.

Finally, as already mentioned, the connection between these logical characterizations and sensing tree automata is a likely place to look for direct connctions between logical and automata-theoretic transductions.

References

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee and Jeffrey Heinz. 2018. Strictly locality and phonological maps. *LI*, 49:23–60.

Jane Chandlee and Adam Jardine. 2019. Quantifier-free least fixed point functions for phonology. In *Proceedings of the 16th Meeting on Mathematics of Language*, pages 50–62. ACL.

Jane Chandlee and Steven Lindell. forthcoming. A logical characterization of strictly local functions. In Jeffrey Heinz, editor, *Doing Computational Phonology*. OUP.

Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. 2008. Tree automata techniques and applications. Published online: http://www.grappa.univ-lille3.fr/tata. Release from November 18, 2008.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *Proceedings of Formal Grammar 2019*, pages 35–50.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

Emmanual Filiot. 2015. Logic-automata connections for transformations. In *Logic and Its Applications (ICLA)*, pages 30–57. Springer.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on Mathematics of Language*, pages 12–26. ACL.

Thomas Graf and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of SCiL 2019*, pages 205–215.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Wim Martens, Frank Neven, and Thomas Schwentick. 2008. Deterministic top-down tree automata: Past, present, and future. In *Proceedings of Logic and Automata 2008*, pages 505–530.

Yukio Otsu. 1980. Some aspects of rendakuin Japanese and related problems. In Ann Farmer and Yukio Otsu, editors, *MIT Working Papers in Linguistics*, pages 207–228. Cambridge, MA: Department of Philosophy and Linguistics, MIT.

Mai Ha Vu, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: A case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 267–276.