

Unit 6

Tree Transducers

Do exercises 6.1–6.3, and either 6.4 or 6.5

Exercise 6.1. The Person Case Constraint (PCC) is a phenomenon in some languages (e.g. French, Catalan, and Slovenian) that forbids certain combinations of direct object (DO) and indirect object (IO) clitics. Which DO-IO combinations are well-formed is contingent on the person specification of the clitics.

- (1) Roger **me/le* *leur* a présenté.
 Roger 1SG/3SG.ACC 3PL.DAT has shown
 ‘Roger has shown me/him to them.’

There are four attested variants:

S(trong)-PCC DO must be 3. [Bonet \(1994\)](#)

U(ltrastrong)-PCC DO is less local than IO, where 3 is less local than 2, and 2 is less local than 1. [Nevins \(2007\)](#)

W(eak)-PCC 3IO combines only with 3DO. [Bonet \(1994\)](#)

M(e first)-PCC If IO is 2 or 3, then DO is not 1. [Nevins \(2007\)](#)

They can be represented more clearly as tables (the diagonal is ignored here, as is commonly done in the literature).

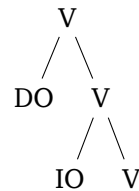
S-PCC				U-PCC			
IO↓/DO→	1	2	3	IO↓/DO→	1	2	3
1	NA	*	✓	1	NA	✓	✓
2	*	NA	✓	2	*	NA	✓
3	*	*	NA	3	*	*	NA
W-PCC				M-PCC			
IO↓/DO→	1	2	3	IO↓/DO→	1	2	3
1	NA	✓	✓	1	NA	✓	✓
2	✓	NA	✓	2	*	NA	✓
3	*	*	NA	3	*	✓	NA

Table 6.1: Attested variants of the PCC

A lot of ink has been spilled on complicated proposals that try to provide syntax with a mechanism to restrict the co-occurrence of DO and IO clitics like this and to

explain why some languages have a PCC. But in fact the existence of PCCs is entirely unsurprising from a computational perspective: the computational apparatus already has everything it needs to produce PCC effects, so why shouldn't they exist in some language?

Assuming the structure below for DO-IO clitic combinations, give a bottom-up tree automaton that describes one of the four PCC variants.



Is it difficult to modify your automaton so that it describes some other PCC variant instead? What does this imply for the interplay of theory and data? \odot

Exercise 6.2. A lot of recent work in Minimalism worries about the status of labels, in particular whether interior nodes in syntactic trees have any labels. Write a linear tree transducer that maps X' -trees (i.e. trees where every phrase has a head) to their label-free counterparts. Then write a linear tree transducer that does the reverse, mapping label-free trees to the corresponding labeled trees. Assume that

- lexical heads have features that encode their category and their subcategorization properties,
- every lexical item of category X projects exactly one X^0 , exactly one X' , and at least one XP ,
- the heads of adjuncts have a feature that indicates their adjunct status,
- adjuncts can adjoin to adjuncts (as in *very old man*).

What does the existence of such automatic translation procedures imply for syntactic research on labels? \odot

Exercise 6.3. We have seen that displacement can be captured without movement by directly merging phrases in their final landing site. We did this with a cascade of transducers:

- a non-deterministic relabeling that freely inserts movement licenser (+wh,+top,...) and licensee nodes (-wh,-top,...),
- a transducer that filters out trees where a $-f$ mover is in the specifier of a head that does not have a $+f$ feature,
- a transducer that filters out trees where a $-f$ trace is in a position that no $-f$ mover could ever originate from (e.g. because the trace is the complement of vP but is specified as -wh)
- a transducer whose states consist of two components:
 - a 0/1-vector where each position is associated with a specific licensee feature (so that $\langle 0, 1, 1 \rangle$, for instance, encodes the presence of a topic mover and a case mover, but the absence of a wh-mover),

- the licensee feature of the current node, if there is one.

Suppose that we want to also enforce some strong island constraints, e.g. the Coordinate Structure Constraint and the Adjunct Island Constraint. Where in this cascade should island constraints be enforced? Can you give an automaton or transducer for the Adjunct Island Constraint? What about the coordinate structure constraint? ☉

Exercise 6.4. In English, wh-movement always induces *do*-support unless the moving phrase is a subject. Building on the movement model of the previous exercise, give a transducer that rewrites C-heads as ϵ or *do* depending on the presence of wh-movement. You may ignore the agreement relation between *do* and the subject. ☉

Exercise 6.5. Affix hopping is a particular analyses of the inflection of finite verbs in English. The claim is that the agreement marker resides in T, where it enters a local agreement relation with the subject in Spec,TP. The marker then affix hops downwards onto the verb. So *John likes Mary* is underlyingly *John -s like Mary*. If affix hopping is blocked due to an intervening head, the affix is instead realized as an inflected form of *do*.

- (2) a. John likes Mary.
 b. * John not likes Mary.
 c. John does not like Mary.

Write a linear bottom-up tree transducer (or a cascade of such transducers) that correctly implements affix hopping. Does the result work similar to the original idea behind affix hopping? ☉