Second position clitics and monadic second-order transduction

Neil Ashton
203 Morrill Hall
Cornell University
Ithaca, NY 14853-4701
nma38@cornell.edu

Abstract

The simultaneously phonological and syntactic grammar of second position clitics is an instance of the broader problem of applying constraints across multiple levels of linguistic analysis. Syntax frameworks extended with simple tree transductions can make efficient use of these necessary additional forms of structure. An analysis of Sahidic Coptic second position clitics in a context-free grammar extended by a monadic second-order transduction exemplifies this approach.

1 Introduction

Second position (2P) clitics are ubiquitous in the world's languages, found in genetically and typologically diverse languages (e.g. Serbo-Croatian, Warlpiri, O'odham) from all documented periods (e.g. Hittite, spoken ca. 1600–1300 BC). They present a persistent challenge for syntactic analysis, inducing a peculiar form of crossing dependency which is not easily expressed in any standard restrictive grammar framework.

2P clitics are emblematic of a wider class of problematic phenomena which existing frameworks can address by incorporating a notion of prosodic constituency. The transductive perspective on mildly context-sensitive grammar formalisms, which treats them as monadic second-order transductions of regular tree languages, suggests how this can be done: by transducing prosodic constituency from syntactic phrase structure.

The prosodic conditioning of 2P clisis is particularly salient in Sahidic Coptic (Reintges, 2004).[1] A context-free phrase structure grammar extended by a monadic second-order transduction is able to make use of the phonological structure necessary to give a linguistically plausible analysis to a fragment of Coptic clitic syntax.

2 Second position clitics and prosodic constituency

2.1 Second position

An intuitive account of the syntax of 2P clitics[2] has been known since Wackernagel (1892). The 2P clitic, which is an immediate functional dependent of a clause, e.g. a sentential adverb, discourse particle, pronominal argument, etc., appears after the *first word* of that clause, potentially interrupting whatever constituent contains that word as its leftmost member, as the chain of 2P clitics interrupts the NP in the following Serbo-Croatian sentence.[3]

(1)	[*Taj*	*=joj=ga=je*	*čovek*]$_{NP}$	*poklonio.*
	that	=her=it=AUX	man	presented

‘That man presented her with it.’ (Bögel et al., 2010)

[1] “Coptic” refers to the latest form of the Egyptian language. Sahidic Coptic, the major literary dialect of Coptic from the 4th to the 10th centuries AD, is survived by a rich corpus of Greek-alphabet texts. The only extant computational model of Sahidic Coptic grammar is apparently that of Orlandi (2004). This work is unfortunately not available to the author, and so no comparison of approaches has been possible.

[2] A “clitic” is, descriptively, a word-like element with affix-like phonological dependence (“clisis”) on other words. Proclitics and enclitics are dependent on right- and left-adjacent words, respectively, and 2P clitics are a special case of enclitics. For more on clitics, see Zwicky (1977), Aikhenvald (2003), and Anderson (2005).

[3] Clitic boundaries are marked with an equals sign, after the Leipzig glossing conventions.

Proc. EACL 2012 Workshop on Applications of Tree Automata Techniques in Natural Language Processing, pages 31–41,
Avignon, France, April 24 2012. ©2012 Association for Computational Linguistics

This constituency-breaking word order pattern alone poses a descriptive challenge. The difficulty is exacerbated by the fact that the "word" targeted by the 2P clitic is not in general syntactically characterizable. It is rather a phonological constituent that may include incorporated clitics (Inkelas and Zec, 1990; Zec, 2005). The alternation in the position of the 2P clitic *de* in the Coptic sentences (2) and (3) illustrates this well.

(2) *a=t=ef=sone* <u>*=de*</u> *ol*
AUX.PF=the=3SG=sister =and gather
en=n=ef=kees
ACC=the=3SG=bones
'and his sister gathered his bones' (Mena, Martyrd. 4a:1-2)

(3) *a=w=tamio* <u>*=de*</u> *en=u=taive*
AUX.PF=3PL=make =and ACC=a=coffin
'and they made a coffin' (Mena, Martyrd. 5a:27-28)

In both sentences, *de* functions as a clausal conjunction. But its position varies, appearing between the main verb and its subject in (2) and between the verb and its object in (3). This alternation is most plausibly phonological. The 2P clitic appears after the first independently pronounceable word, including its attached clitics, such as the pronominal subject *w-* in (3) and the tense auxiliary *a-* in both sentences. The behavior of 2P clitics when the verb itself or its direct object are clitics is consistent with this analysis.

Phonological properties alone, however, do not suffice to describe the syntax of 2P clitics. They are constrained to appear within a syntactically determined subpart of their host clause, typically ignoring topicalized or otherwise left-dislocated elements and thus appearing quite far from strict phonological second position. Describing 2P clisis thus requires reference to both syntactic and phonological structure.

2.2 Prosodic constituency via tree transduction

The notion of prosodic constituency (Nespor and Vogel, 1986; Selkirk, 1986) provides the key to a perspicuous account of the multiple factors at play in the grammar of 2P clitics. Prosodic constituency is a tree structure that defines the "words" and "phrases" relevant to phonology, which are in general distinct from yet closely related to their syntactic equivalents.

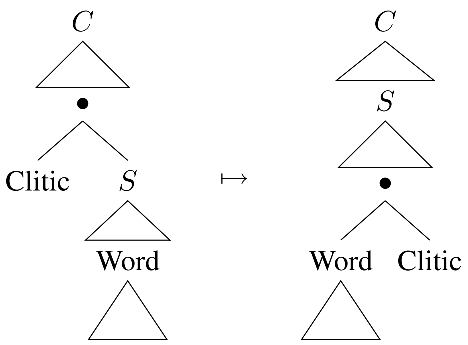

Figure 1: Lowering 2P clitics.

Both the distinctness of and the relationship between syntactic and prosodic constituency can be captured by transducing the latter from the former. This transduction in effect interprets syntactic trees as terms over a signature of phonological operations and normalizes the result. The yield function is a prosodically naive example of such a transduction.

Once this independently necessary transduction has been taken into account, the syntax of 2P clitics is straightforward. The 2P clitic simply has a non-concatenative mode of phonological combination. The clitic and its host clause are siblings in syntactic constituency, and their parent node is interpreted as an operation that wraps the latter around the former—alternatively, lowers the former into the latter.

This analysis, which captures in essence both the "wrapping" (Bach, 1987) and "prosodic inversion" (Halpern, 1995) analyses of 2P clitics, can be schematized as in Figure 1, where "Word" is constrained to be the leftmost node with that label in S.

This transduction is not direction-preserving in the sense of Bloem and Engelfriet (2000): assuming that the clitic crosses unboundedly many nodes on the way to its host word, a crossing dependency is induced in the paths of the target tree. This rules out the possibility of formalizing this analysis by means of popular automaton models such as multi bottom-up tree transducers (Fülöp et al., 2004) or their extended variant (Engelfriet et al., 2009), which cannot describe such dependencies (Maletti, 2011).

The more powerful automata that can be specified using monadic second-order logic (MSO), which include syntactically restricted classes of macro tree transducers (Engelfriet and Maneth, 1999) and deterministic tree-walking transducers (Bloem and Engelfriet, 2000), can perform this transduction. Section 3 defines the transduction in MSO, and Section 4 reflects briefly on its implementation.

3 Sahidic Coptic 2P clitics via CFG+MST

The following context-free grammar and sequence of MSO transductions formalizes, for a fragment of Sahidic Coptic, the analysis of 2P clisis sketched in Section 2.2.

Section 3 breaks the interpretation of a syntactic parse tree as a phonological term into a series ($f_1 - f_7$) of simple composed MSO transductions. A "redex" phonological term is derived (Section 3.3), and its reducible subterms are then evaluated separately (Section 3.4). An algorithmic implementation of the transduction is sketched in Section 3.5.

3.1 Formal preliminaries

The following definitions and assertions rehearse material from Courcelle and Engelfriet (2012), which should be consulted for full details.

3.1.1 Relational structures and tree graphs

A relational signature is a finite set R of relation symbols with associated arity $\rho(r) \in \mathbb{N}^*$ for each $r \in R$. A relational structure over R is a tuple $\mathscr{R} = \langle D_{\mathscr{R}}, (r_{\mathscr{R}})_{r \in R} \rangle$, where $D_{\mathscr{R}}$ is a finite domain of entities and $r_{\mathscr{R}}$, for each $r \in R$, is a $\rho(r)$-ary relation on $D_{\mathscr{R}}$.

A bijection exists between binary relational structures and labelled graphs, with unary and binary relations corresponding to node and edge labels, respectively. Ordered binary trees can be represented as labelled directed graphs, and hence as relational structures, in the obvious way.

3.1.2 Monadic second-order logic

The monadic second-order (MSO) formulas over a relational signature R are as first-order predicate logic, with the addition of monadic second-order variables $X, Y, X', \ldots$ denoting sets of entities, second-order quantification, and a primitive operator for set membership. The substitution of n free variables in a formula ϕ by entities $d_1, \ldots, d_n$ is written $\phi(d_1, \ldots, d_n)$.

An MSO formula over R is interpreted in a relational signature over R. A formula with no free variables is called a sentence. If a sentence ψ is true in a relational structure $\mathscr{R}$, we write $\mathscr{R} \models \psi$, pronounced "$\mathscr{R}$ models ψ".

3.1.3 MSO transduction

An MSO transduction defines a relational structure in terms of another by taking a finite number of copies of nodes from the source domain, keeping those that satisfy particular formulas in the source structure, and defining the relations that hold in the target structure by means of formulas modeled by the source structure. The generalization of MSO transduction to k-copying MSO transduction (Courcelle, 1991) allows the target domain to be larger than its source. MSO transductions whose formulas do not refer to parameters define deterministic functions.

A (parameterless, k-copying) MSO transduction over a relational signature R is specified by a triple $\langle k, \Delta, \Theta \rangle$, where $k \in \mathbb{N}$ and $\Delta = \{\delta_i \mid 0 \leq i \leq k\}$ and $\Theta = \{\theta_w \mid w \in W\}$ are sets of MSO formulas with free variables, and W is the set of all tuples $(r, i_1, \ldots, i_{\rho(r)})$ for $r \in R$. This triple is called a definition scheme.

A definition scheme specifies a target relational structure T with respect to a source relational structure S as follows. The domain D_T of T is the set $(D_0 \times \{0\}) \cup \ldots \cup (D_k \times \{k\})$, where each $D_i = \{d \in D_S \mid S \models \delta_i(d)\}$. For each n-ary relation r in the relational signature of T, an n-ary relation on D_T is defined as:

$$\bigcup_{i_0, \ldots, i_n \in [k]} \begin{array}{l} \{((d_0, i_0), \ldots, (d_n, i_n)) \mid \\ \quad d_0 \in D_{i_0}, \ldots, d_n \in D_{i_n}, \\ \quad S \models \theta_{r, i_0, \ldots, i_n}(d_0, \ldots, d_n)\} \end{array}$$

Intuitively, a formula δ_i specifies conditions on the existence of the ith copy of a node in the target structure. A formula $\theta_{(r, i_0, \ldots, j_{\rho(r)})}$ specifies conditions on the relation r holding between copies of nodes indexed $i, \ldots, j$ in the target structure.

3.2 Definitions and abbreviations

3.2.1 Base CFG

The phrase structure grammar which serves as the basis of the analysis of Coptic is presented in

$S \rightarrow Cl\ S'$
$S' \rightarrow Aux\ VP$
$VP \rightarrow NP_N\ V'$
$VP \rightarrow NP_{pro}\ V'$
$V' \rightarrow V\ AccP$
$Cl \rightarrow$ de
$Aux \rightarrow$ a
$V \rightarrow$ ol | tamio
$N^{sg}_{fem} \rightarrow$ sone | taive
$N^{pl} \rightarrow$ kees
$Acc_N \rightarrow$ en
$Acc_{pro} \rightarrow$ mmo
$NP_{pro} \rightarrow Pro$
$NP_N \rightarrow Det^{sg}_{fem}\ N'^{sg}_{fem}$
$NP_N \rightarrow Det_{indef}\ N^{sg}_{fem}$
$NP_N \rightarrow Det^{pl}\ N'^{pl}$
$N'^{sg}_{fem} \rightarrow NP_{pro}\ N^{sg}_{fem}$
$N'^{pl} \rightarrow NP_{pro}\ N^{pl}$
$AccP \rightarrow Acc_N\ NP_N$
$AccP \rightarrow Acc_{pro}\ NP_{pro}$
$Det^{sg}_{fem} \rightarrow$ t
$Det^{pl} \rightarrow$ n
$Det_{indef} \rightarrow$ u
$Pro \rightarrow$ w | ef

Figure 2: Base CFG fragment of Coptic.

Figure 2. Its parse trees define a recognizable language of binary trees, members of which can be represented as relational structures, as explained in Section 3.1.1. This CFG fragment, in combination with the transductions detailed below, suffices to generate sentences (2) and (3) from Section 2.1.

This grammar encodes several claims, already alluded to in Section 2.1, about the syntactic structure of Coptic. Syntactic dependencies are represented by constituency in the usual way. The immediate dependence of the 2P clitic *de* on a host clause is expressed by the siblinghood of *Cl* and *S′* under *S*.

Features of lexical items relevant for agreement and allomorphy are encoded as diacritics on nonterminals, allowing determiners to agree with nouns in gender and the accusative case preposition to covary with the nominal or pronominal status of its complement.

3.2.2 Encoding of nodes with unbounded branching

Syntactic trees are interpreted into prosodic trees, which may contain prosodic word constituents that branch unboundedly wide. To fix a binary encoding for such constituents, a "*cons* cell"-like variant of the extension operator encoding (Comon et al., 2007, p. 210) is adopted, in which a term of the form $@(x, y)$ is interpreted as extending the interpretation of y by adding x to its root as its leftmost child. An example of this encoding is given in Figure 3.

Only the fragment of prosodic constituency relevant to the alternation shown in sentences (2) and (3) is derived. The output tree therefore contains operator-encoded prosodic constituents as subtrees of *unencoded* trees containing unanalyzed phonological combination operators.

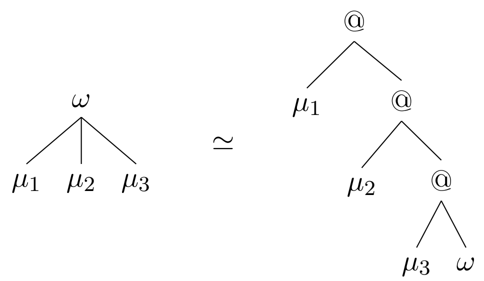

Figure 3: Encoding of n-ary trees.

$\frown$	concatenation
$\bullet_p$	proclisis
$\bullet_e$	enclisis
$\bullet_{2p}$	2P clisis
$\bullet_{id}$	identity
ω	prosodic word
$@$	extension operator

Table 1: Interpretation of labels.

3.2.3 Relational signature and abbreviations

All MSO transductions presented below are defined over a binary relational signature $R = R_1 \cup R_2$. The set of node labels R_1 is given by the union of the set of all non-terminal and terminal node names in the grammar of Figure 2 and the set $\{\frown, \bullet_p, \bullet_e, \bullet_{2p}, \bullet_{id}, \omega, @\}$. The interpretation of these predicates is given in Table 1. The set of binary predicates R_2 is simply $\{\lhd_0, \lhd_1\}$, the left and right child relations, written as infix operators as a notational convenience.

It will be useful to define several new binary predicates as syntactic abbreviations. I assume reflexive and irreflexive transitive closures r^* and r^+ of relations $r \in R_2$, as well as immediate domination and precedence $\lhd, \prec$, as abbreviations of MSO formulas over primitive predicates.[4]

Recurring structural properties of lexical items in the base CFG are given by the unary syntactic abbreviations defined below.[5] These include pro-

[4] On the MSO-definability of these, see Courcelle and Engelfriet (2012).

[5] "$\psi := \phi$" is to be read "ψ is an abbreviation for ϕ".

clitic and 2P clitic status ($\mathit{Pc}(x)$, $\mathit{2P}(x)$), independent pronounceability ($\mathit{Str}(x)$), and the property of being a leaf ($\mathit{Leaf}(x)$).

$$\begin{aligned}\mathit{Pc}(x) &:= \mathrm{a}(x) \vee \mathrm{en}(x) \vee \mathrm{t}(x) \vee \mathrm{n}(x)\\ \mathit{2P}(x) &:= \mathrm{de}(x)\\ \mathit{Str}(x) &:= \mathrm{ol}(x) \vee \mathrm{sone}(x)\\ &\qquad \vee \mathrm{kees}(x) \vee \mathrm{mmo}(x)\\ \mathit{Leaf}(x) &:= \mathrm{de}(x) \vee \mathrm{a}(x) \vee \ldots\end{aligned}$$

MSO transductions are given by transduction schemes, as defined in Section 3.1.3. In the case that $k = 0$, irrelevant subscripts are omitted. Unless otherwise specified, all formulas δ_i can be assumed to be the constant *True*.

3.3 Transducing a reducible term

A syntactic constituency tree can be interpreted as a term in a phonological algebra, with non-leaf nodes interpreted as operations effecting phonological combination in various modes. Pronounceable utterances, which consist of concatenations of prosodic constituents (i.e. terms over leaves from the base CFG, @, ω, and $\frown$), are normal forms.

This complex interpretation is broken into smaller transductions, the first set of which lays the foundation for the reduction of the "clitic" modes of combination. Non-leaf nodes are first replaced by appropriate combination operators (Section 3.3.1). Unary nodes are then eliminated (Section 3.3.2). Finally, the prosodic structure necessary for the next phase of interpretation is generated (Section 3.3.3).

3.3.1 Relabeling

Non-terminal leaves in the syntactic tree are replaced by operators indicating modes of phonological combination, as presented in Table 1.

The transduction to unreduced phonological terms is sensitive to the structure of the syntactic tree. Some leaves, e.g. clitic pronouns, are not strictly proclitic or enclitic but vary by context: the pronominal subject of a verb or possessor of a noun is proclitic, whereas the pronominal complement of an accusative preposition or pronoun-selecting verb is enclitic. The relevant syntactic context is the child status of NP_{pro} nodes. Hence the parent of an NP_{pro} node is replaced by $\bullet_p$ if NP_{pro} is its left child, by $\bullet_e$ if NP_{pro} its right child.

All non-pronominal clitics are phonologically combined with the sibling of their phonologically vacuous unary parent node. Thus the grandparents of all such clitic leaves are replaced by the appropriate clitic combination operator, $\bullet_p$ for proclitics and $\bullet_{2p}$ for 2P clitics. Unary nodes are replaced by $\bullet_{id}$, and all other non-leaf nodes are replaced by $\frown$. Leaf node labels are left unchanged.

The definition scheme $f_1 = \langle 0, \Delta, \Theta \rangle$, where Θ is defined as the union of the formulas given below, specifies this transduction. The body of the $\theta_\frown$ formula, which consists largely of the disjunction of the negations of the preceding formulas, is omitted, as signaled by $[etc]$; and the θ_w formula which reasserts leaf labels is omitted altogether.

$$\begin{aligned}\theta_{\bullet_e}(x) &= \exists x'(\mathit{NP}_{pro}(x') \wedge x \lhd_1 x')\\ \theta_{\bullet_p}(x) &= \exists x'(\mathit{NP}_{pro}(x') \wedge x \lhd_0 x')\\ &\quad \vee \exists x', x''(x \lhd_0 x' \wedge x' \lhd_0 x'' \wedge \mathit{Pc}(x''))\\ \theta_{\bullet_{2p}}(x) &= \exists x', x''(x \lhd_0 x' \wedge x' \lhd_0 x'' \wedge 2P(x''))\\ \theta_{\bullet_{id}}(x) &= \exists x'(x \lhd_0 x') \wedge \neg\exists x''(x \lhd_1 x'')\\ \theta_\frown(x) &= [etc]\end{aligned}$$

3.3.2 Eliminating unary nodes

Before any further interpretation takes place, unary $\bullet_{id}$ nodes, which are phonologically vacuous, can be eliminated.

The definition scheme $f_2 = \langle 0, \Delta, \Theta \rangle$, with Θ defined as the union of the following formulas (for $i \in \{0, 1\}$), eliminates unary nodes by connecting a non-$\bullet_{id}$ node dominated by a path of $\bullet_{id}$ nodes to the parent of the topmost $\bullet_{id}$ in the path. Again, $[etc]$ stands for the omitted "elsewhere condition", which here reasserts edges from the source.

$$\begin{aligned}\theta_{\lhd_i}(x, y) = {}&\neg \bullet_{id}(x) \wedge \neg \bullet_{id}(y)\\ &\wedge \exists x'(x \lhd_i x' \wedge x' \lhd^+ y\\ &\quad \wedge \forall y'(x' \lhd^* y' \wedge y' \lhd^+ y\\ &\qquad \rightarrow \bullet_{id}(y'))) \vee [etc]\end{aligned}$$

An example of the composed transduction $f_2 \circ f_1$ is given in Figure 4.

3.3.3 Base prosodic words

Before reducing the remaining reducible modes of combination, it is necessary to create prosodic word constituents, notated ω, that cover the independently pronounceable "strong" leaves of the tree, allowing the word-sensitive clitic modes of combination to be interpreted correctly. Prosodic

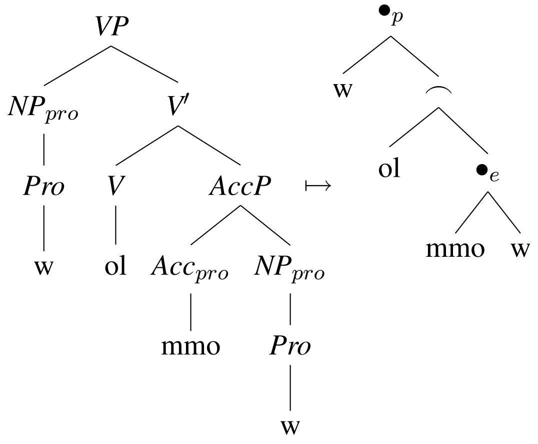

Figure 4: Relabeling and $\bullet_{id}$-elimination.

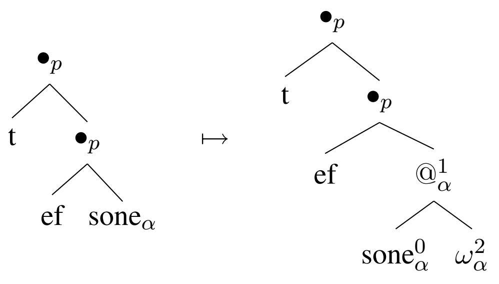

Figure 5: Prosodic word insertion.

words are encoded by the scheme given in Section 3.2.

The definition scheme $f_3 = \langle 2, \Delta, \Theta \rangle$, with Δ and Θ the union of the δ and θ formulas below, specifies a transduction that takes two additional copies of all nodes, relabels the copies of strong leaf nodes as @ and ω, and draws edges as appropriate.

$$
\begin{aligned}
\delta_1(x) = \delta_2(x) &= \\
\theta_{(@,1)}(x) = \theta_{(\omega,2)}(x) &= \mathit{Str}(x) \\
\theta_{(\lhd_1,0,0)}(x,y) &= \neg \mathit{Str}(y) \wedge x \lhd_1 y \\
\theta_{(\lhd_1,0,1)}(x,y) &= \mathit{Str}(y) \wedge x \lhd_1 y \\
\theta_{(\lhd_0,1,0)}(x,y) &= \\
\theta_{(\lhd_1,1,2)}(x,y) &= \mathit{Str}(x) \wedge x = y \\
\theta_{(\lhd_0,0,0)}(x,y) &= \mathit{True}
\end{aligned}
$$

An example of the tree transduction given by f_3 is shown in Figure 5, with identity of copies indicated by subscript letters and the number of the copy by superscript numerals.

3.4 Interpreting clitic combination modes

The composed transduction $f_3 \circ f_2 \circ f_1$ produces reducible phonological terms in which the prosodic structure necessary to interpret the clitic modes of combination ($\bullet_p$, $\bullet_e$, and $\bullet_{2p}$) is present.

The interpretation of the clitic modes proceeds in three steps. "Local" clitics, siblings of prosodic words, are amalgamated into their hosts (Section 3.4.1). "Long-distance" clitics, which are not thus locally attached, are lowered to their hosts (Section 3.4.2) and then attached as local clitics. Second-position clitics are finally lowered and attached by the same means, as a special case (Section 3.4.3).

3.4.1 Local clisis

Locally connected clitics can be directly incorporated into their hosts. The word constituent so derived is the recursive structure (e.g. $[_\omega clitic\ [_\omega host]]$) generally assumed for cliticized words (cf. Inkelas and Zec, 1990; Zec, 2005).

Proclitics and enclitics can be interpreted separately. For proclitics, the relevant notion of "locality" can be expressed by a predicate $\circ_p(x)$, which identifies $\bullet_p$ nodes connected to @ nodes by a path of $\bullet_p$ nodes.

$$
\begin{aligned}
\circ_p(x) := \ \bullet_p\,(x) \wedge \exists y (@(y) \\
\wedge\, x \lhd_1^+ y \wedge \forall z (x \lhd^* z \\
\wedge\, z \lhd^+ y \rightarrow \bullet_p(x)))
\end{aligned}
$$

The 2-copying MS transduction specified by the definition scheme $f_4 = \langle 2, \Delta, \Theta \rangle$, with Δ and Θ given by the union of the δ and θ formulas below, produces the appropriate bracketing by projecting a new word above each proclitic and relocating each proclitic's sibling to the new word constituent.

$$
\begin{aligned}
\delta_1(x) = \delta_2(x) = \theta_{(@,0)}(x) &= \\
\theta_{(@,1)}(x) = \theta_{(\omega,2)}(x) &= \ \circ_p\,(x) \\
\theta_{(\lhd_1,0,1)}(x,y) = \theta_{(\lhd_1,1,2)}(x,y) &= \ \circ_p\,(x) \wedge x = y \\
\theta_{(\lhd_0,1,0)}(x,y) &= \ \circ_p\,(x) \wedge x \lhd_1 y \\
\theta_{(\lhd_0,0,0)}(x,y) = \theta_{(\lhd_1,0,0)}(x,y) &= [etc]
\end{aligned}
$$

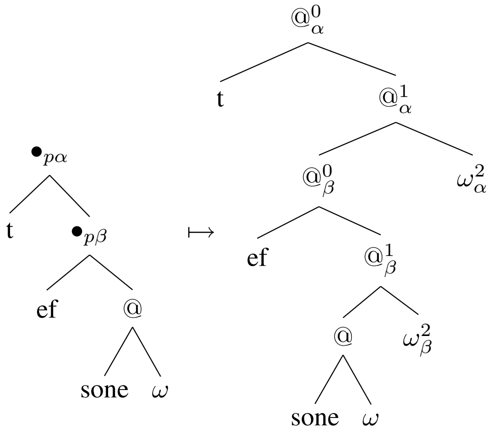

Figure 6: Local proclisis.

Figure 6 gives an example of a tree transformation effected by f_4, again with subscripts and superscripts indicating copies.

The interpretation of local enclitics proceeds similarly. A predicate $\circ_e(x)$ defines the relevant notion of locality.

$$\begin{aligned}\circ_e(x) :=\ & \bullet_e\,(x) \wedge \exists y(@(y) \\ & \wedge x \lhd_0^+ y \wedge \forall z(x \lhd^* z \\ & \quad \wedge z \lhd^+ y \to \bullet_e(x)))\end{aligned}$$

The transduction $f_5 = \langle 2, \Delta, \Theta\rangle$, with Δ and Θ given by the union of the δ and θ formulas below, produces the appropriate bracketing. This transduction is more complicated than the proclitic transformation in that enclitics, right children in the source tree, must be relocated to left branches of @ nodes.

$$\begin{aligned}\delta_1(x) = \delta_2(x) &= \\ \theta_{(@,0)}(x) = \theta_{(@,1)}(x) &= \\ \theta_{(\omega,2)}(x) &= \circ_e\,(x) \\ \theta_{(\lhd_0,1,0)}(x,y) &= \circ_e\,(x) \wedge x \lhd_0 y \\ \theta_{(\lhd_1,1,0)}(x,y) &= \\ \theta_{(\lhd_1,0,2)}(x,y) &= \circ_e\,(x) \wedge x = y \\ \theta_{(\lhd_0,0,0)}(x,y) &= \circ_e\,(x) \wedge x \lhd_1 y \vee [etc] \\ \theta_{(\lhd_1,0,0)}(x,y) &= [etc]\end{aligned}$$

Figure 7 gives an example of the tree transduction specified by f_5.

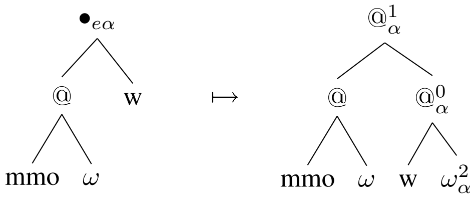

Figure 7: Local enclisis.

3.4.2 Long-distance proclisis

Long-distance clitics, which are not locally combined with their hosts, incorporate into them in the same manner as local clitics (i.e. by transductions f_4 and f_5) but must be lowered to them to do so.

Only long-distance proclisis is relevant to the grammar fragment under consideration. A long-distance proclitic is a non-local proclitic (see Section 3.4.1 for the notion of "locality") adjacent to a word in the yield, ignoring other proclitics. Pronouns count as proclitics for this purpose, so a predicate $Pc'(x)$ including pronouns is defined. The predicate $Adj(x, y)$ expresses adjacency of x and y, and the predicate $L_p(x)$, which identifies the parents of long-distance proclitics, is defined in terms of $Adj(x, y)$.

$$\begin{aligned}Pc'(x) &:= Pc(x) \vee \mathrm{w}(x) \vee \mathrm{ef}(x) \\ Adj(x,y) &:= x \prec y \wedge \forall x'(x \prec x' \\ &\qquad \wedge x' \prec y \wedge Leaf(x') \\ &\qquad \to Pc'(x')) \\ L_p(x) &:=\ \bullet_p\,(x) \wedge \exists x', y(@(y) \\ &\qquad \wedge x \lhd_0 x' \wedge Adj(x', y))\end{aligned}$$

The parents of long-distance proclitics get attached to "goal" nodes—that is, @ nodes or other parents of long-distance proclitics—by the right child relation. The predicate $G(x)$ identifies goals, and $NG(x, y)$ identifies node x's nearest goal y.

$$\begin{aligned}G(x) &:=\ \bullet_p\,(x) \vee @(x) \\ NG(x,y) &:= x \lhd^+ y \wedge G(y) \wedge \forall y'(x \lhd^+ y' \\ &\qquad \wedge G(y') \to y \prec y')\end{aligned}$$

The parent of the topmost in a path of $\bullet_p$ nodes must get attached, by whatever child relation connects that parent node to that path, to the right child of the lowest node in the path. The higher-order syntactic abbreviation $PC[i;x,y]$ specifies the relevant relation, whereby a path of $\bullet_p$ nodes begins with the ith child of x and leads to y.

$$\begin{aligned} PC[i;x,y] := {} & \neg \bullet_p (x) \wedge \neg \bullet_p (y) \\ & \wedge \exists x'(\bullet_p(x') \wedge x \lhd_i x' \\ & \wedge x' \lhd_1^+ y \wedge \forall y'(x' \lhd^* y' \\ & \wedge y' \lhd_1^+ y \rightarrow \bullet_p(y'))) \end{aligned}$$

The parent of a @ node targeted by a set of long-distance clitics gets attached to the highest parent of a clitic in that set. The predicate $Hi_p(x)$ identifies such highest proclitic parents. Only "maximal" @ nodes, those that are highest in the right-recursive path of @ nodes leading to an ω, are relevant; these are identified by the predicate $Max@p(x)$. The abbreviation $WC[i;x,y]$ identifies a highest $\bullet_p$ node y adjacent to a maximal @ node that is the ith child of x.

$$\begin{aligned} Hi_p(x) := {} & L_p(x) \wedge \exists x'(x \lhd_0 x' \\ & \wedge \forall y(y \prec x' \rightarrow \neg Pc'(y))) \\ Max@p(x) := {} & @(x) \wedge \neg\exists y(y \lhd_1 x \wedge @(y)) \\ & \wedge \exists z(x \lhd_1^+ z \wedge \omega(z)) \\ WC[i;x,y] := {} & \exists x', y'(Max@p(x') \\ & \wedge x \lhd_i x' \wedge y \lhd_0 y' \\ & \wedge Adj(y', x') \wedge Hi_p(y)) \end{aligned}$$

Once these auxiliary predicates are defined, a simple MSO transduction $f_6 = \langle 0, \Delta, \Theta \rangle$ meeting the specifications given above can be defined by the union of the following formulas.

$$\begin{aligned} \theta_{\lhd_1}(x,y) = {} & \bullet_p (x) \wedge NG(x,y) \\ & \vee PC[1;x,y] \vee WC[1;x,y] \vee [etc] \\ \theta_{\lhd_0}(x,y) = {} & PC[0;x,y] \vee WC[0;x,y] \vee [etc] \end{aligned}$$

Figure 8 gives an example of the transduction specified by f_6. The transduction f_4 can be composed with f_6 to produce the appropriate constituency for the lowered proclitics.

3.4.3 Second-position clisis

There is little substantive difference between long-distance proclitics and 2P clitics—both arrive in their position by a "lowering" transformation that targets @ nodes. The transductions already defined can be recycled, essentially unchanged, to derive 2P clisis.

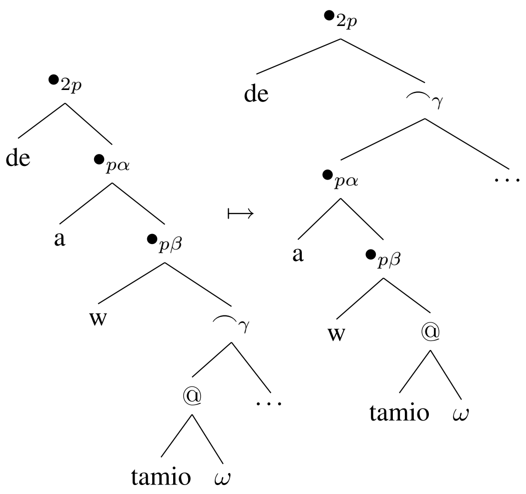

Figure 8: Long-distance proclisis, part 1: lowering.

Assume a lowering transduction f_6' identical to f_6 except operating on $\bullet_{2p}$ nodes. The resulting lowered 2P clitics, which are in a "proclitic" configuration, can then be "rotated" and relabeled as enclitics. The MSO transduction $f_7 = \langle 0, \Delta, \Theta \rangle$ given by the union of the following formulas produces this transformation.

$$\begin{aligned} \theta_{\bullet_e}(x) = {} & \bullet_{2p} (x) \\ \theta_{\lhd_0}(x,y) = {} & \neg \bullet_{2p} (x) \wedge x \lhd_0 y \\ & \vee \bullet_{2p}(x) \wedge x \lhd_1 y \\ \theta_{\lhd_1}(x,y) = {} & \neg \bullet_{2p} (x) \wedge x \lhd_1 y \\ & \vee \bullet_{2p}(x) \wedge x \lhd_0 y \end{aligned}$$

The local enclisis transduction f_5 is then applied to incorporate the 2P clitics into their hosts. An example transformation effected by the transduction $f_5 \circ f_7 \circ f_6'$ is shown in Figure 3.4.3.

3.5 Algorithmic implementation

No automaton compiler for MSO transductions exists, and the non-elementary complexity of the MSO-to-automaton translation procedure ensures that the development of a practical compiler will be a difficult undertaking. The most convenient algorithmic implementation of the above analysis is therefore an indirect one: an extensionally equivalent algorithm constructed in an expressively equivalent transduction framework.

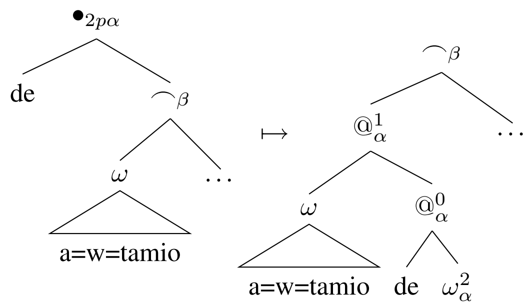

Figure 9: Second position clisis.

Second-order Abstract Categorial Grammar (Kanazawa, 2009b) is one such framework, equivalent to MSO in tree-transforming power (Kanazawa, 2009a). ACG tree transductions, which are expressed as linear λ-term homomorphisms and thus have the same complexity as linear λ-term normalization, can be implemented in Haskell in the manner of Kiselyov and Shan (2010). A function extensionally equivalent to that defined logically above can be defined in a simple ACG consisting of a composed pair of homomorphisms and implemented in Haskell in a pair of type classes.

4 Discussion and conclusion

The analysis of Sahidic Coptic 2P clitics in terms of prosodic constituency and tree transformation given above successfully accounts for the alternation shown in sentences (2) and (3). It promises to scale to a larger fragment of Coptic grammar, accommodating the addition of clitic main verbs and direct objects without further ado. The general approach also promises to extend straightforwardly to other languages with 2P clitics, such as Russian and Hittite. Since the general technique of MSO transduction underlying the analysis applies to all tree-deriving grammar formalisms, richer grammatical backbones than CFG can be deployed as necessary.

This transductive analysis is in line with a nascent convergence in perspectives on restrictive formal syntax. The mildly context-sensitive languages, polynomially parseable languages containing limited cross-serial dependencies such as those induced by 2P clitics, have received a new logical characterization in light of the past decade's surge of interest in disentangling derivations from their interpretations.[6] Mildly context-sensitive languages are the images of recognizable tree languages under monadic second-order transductions.[7] This generalizes not only string-generating formalisms like linear context-free rewriting systems (Vijay-Shanker et al., 1987; Weir, 1992) but also context-free languages of graphs (Engelfriet and Maneth, 2000) and linear λ-terms (Kanazawa, 2009a; Kanazawa, 2010).[8]

This perspective suggests a modular approach to framework revision in the face of problematic natural language phenomena. Transductive interpretations are an integral, if not universally recognized, component of restrictive grammar frameworks. Hence, to meet new descriptive challenges such as those posed by 2P clitics, it is natural to extend those frameworks' interpretive components by means of MSO rather than rebuilding them from scratch.

No software toolkit for MSO transduction comparable to the XFST toolkit for regular expressions (Beesley and Karttunen, 2003) or the MONA toolkit for MSO (Henriksen et al., 1995) presently exists, however. Nevertheless, MSO is an excellent candidate for a high-level specification language for tree transformations, promising to play the same role for tree transduction that languages such as XFST play for string transduction. MSO meanwhile serves the useful purpose of providing a denotational check on the complexity of tree transformation algorithms.

Acknowledgments

Many thanks to John Hale, Sarah Murray, Michael Weiss, and three anonymous reviewers for their valuable comments. Thanks also to Chung-chieh Shan, Chris Barker, Greg Kobele, Makoto Kanazawa, and Zoltán Varjú for their conversation and inspiration. This research was supported by the Social Sciences and Humanities Research Council.

[6] See for instance Michaelis et al. (2000), de Groote (2001), Ranta (2002), Morawietz (2003), Muskens (2003), and Pollard (2008), among many others.

[7] See Kolb et al. (2003) for an application of this perspective to the purely syntactic crossing dependencies of Dutch and Swiss German noted by a reviewer.

[8] Closely related perspectives can be found in the frameworks of second-order Abstract Categorial Grammar and Koller & Kuhlmann (2011)'s "interpreted regular tree grammar" paradigm.

References

Alexandra Y. Aikhenvald. 2003. Typological parameters for the study of clitics, with special reference to Tariana. In Robert M. W. Dixon and Alexandra Y. Aikhenvald, editors, *Word: a Cross-Linguistic Typology*, pages 42–78. Cambridge University Press, Cambridge.

Stephen R. Anderson. 2005. *Aspects of the Theory of Clitics*. Oxford University Press, Oxford.

Emmon Bach. 1987. Some generalizations of categorial grammars. In Walter J. Savitch, Emmon Bach, William Marsh, and Gila Safran-Naveh, editors, *The Formal Complexity of Natural Language*, pages 251–279. D. Reidel, Dordrecht.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford.

Roderick Bloem and Joost Engelfriet. 2000. A comparison of tree transductions defined by monadic second order logic and by attribute grammars. *Journal of Computer and System Sciences*, 6(1):1–50.

Tina Bögel, Miriam Butt, Ronald M. Kaplan, Tracy Holloway King, and John T. Maxwell. 2010. Second position and the prosody-syntax interface. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG10 Conference*, pages 107–126.

Hubert Comon, Max Dauchet, Remi Gilleron, Christof Löding, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. 2007. Tree automata techniques and applications. Available at: http://www.grappa.univ-lille3.fr/tata.

Bruno Courcelle and Joost Engelfriet. 2012. Graph structure and monadic second-order logic: a language theoretic approach. In press.

Bruno Courcelle. 1991. The monadic second-order logic of graphs V: on closing the gap between definability and recognizability. *Theoretical Computer Science*, 80:153–202.

Philippe de Groote. 2001. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting*, pages 148–155.

Joost Engelfriet and Sebastian Maneth. 1999. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Information and Computation*, 154:34–91.

Joost Engelfriet and Sebastian Maneth. 2000. Tree languages generated by context-free graph grammars. In Hartmut Ehrig, editor, *Graph Transformation*, pages 15–29, Berlin and Heidelberg. Springer Verlag.

Joost Engelfriet, Eric Lilin, and Andreas Maletti. 2009. Extended multi bottom-up tree transducers: Composition and decomposition. *Acta Informatica*, 46:561–590.

Zoltán Fülöp, Armin Kühnemann, and Heiko Vogler. 2004. A bottom-up characterization of deterministic top-down tree transducers with regular lookahead. *Information Processing Letters*, 91:57–67.

Aaron Halpern. 1995. *On the Placement and Morphology of Clitics*. CSLI Publications, Stanford.

Jesper G. Henriksen, Jakob Jensen, Michael Jørgensen, Nils Klarlund, Robert Paige, Theis Rauhe, and Anders Sandholm. 1995. MONA: Monadic second-order logic in practice. *Lecture Notes in Computer Science*, 1019:89–110.

Sharon Inkelas and Draga Zec. 1990. Prosodically constrained syntax. In Sharon Inkelas and Draga Zec, editors, *The phonology–syntax connection*, pages 365–378. University of Chicago Press, Chicago.

Makoto Kanazawa. 2009a. A lambda calculus characterization of MSO definable tree transductions. Talk given at the 10th Asian Logic Conference.

Makoto Kanazawa. 2009b. Second-order abstract categorial grammars. Manuscript.

Makoto Kanazawa. 2010. Second-order abstract categorial grammars as hyperedge replacement grammars. *Journal of Language, Logic, and Information*, 19(2):137–161.

Oleg Kiselyov and Chung-chieh Shan. 2010. Lambda: the ultimate syntax-semantics interface. NASSLLI 2010 course notes.

Hans-Peter Kolb, Jens Michaelis, Uwe Mönnich, and Frank Morawietz. 2003. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science*, 293:261–289.

Alexander Koller and Marco Kuhlmann. 2011. A generalized view on parsing and translation. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 2–11.

Andreas Maletti. 2011. Tree transformations and dependencies. *Lecture Notes in Computer Science*, 6878:1–20.

Jens Michaelis, Uwe Mönnich, and Frank Morawietz. 2000. Derivational minimalism in two regular and logical steps. In *Proceedings of TAG+ 5*.

Frank Morawietz. 2003. *Two-Step Approaches to Natural Language Formalisms*. Mouton de Gruyter, Berlin and New York.

Reinhard Muskens. 2003. Language, lambdas, and logic. In Richard T. Oehrle and Geert-Jan Kruijff, editors, *Resource sensitivity in binding and anaphora*, pages 23–54. Kluwer, Dordrecht.

Marina Nespor and Irene Vogel. 1986. *Prosodic Phonology*. Foris, Dordrecht.

Tito Orlandi. 2004. Towards a computational grammar of Sahidic Coptic. In Jacques van der Vliet and Mat Immerzeel, editors, *Coptic studies on the threshold of a new millennium*, pages 125–130, Leuven. Peeters.

Carl Pollard. 2008. An introduction to convergent grammar. Manuscript.

Aarne Ranta. 2002. Grammatical Framework. *Journal of Functional Programming*, 14:145–189.

Chris Reintges. 2004. *Coptic Egyptian (Sahidic Dialect).* Rüdiger Köppe Verlag, Köln.

Elisabeth Selkirk. 1986. On derived domains in sentence phonology. *Phonology Yearbook*, 3:371–405.

K. Vijay-Shanker, David J. Weir, and Aravind K. Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*.

Jacob Wackernagel. 1892. Über ein Gesetz der indogermanischen Wortstellung. *Indogermanische Forschungen*, 1:333–436.

David J. Weir. 1992. Linear context-free rewriting systems and deterministic tree-walking transducers. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*.

Draga Zec. 2005. Prosodic differences among function words. *Phonology*, 22:77–112.

Arnold M. Zwicky. 1977. On clitics. Manuscript.