


THE COPY THEORY OF MOVEMENT AND LINEARIZATION OF CHAINS IN THE MINIMALIST PROGRAM

by Jairo Nunes

reviewed by Hans-Martin Gärtner

Summary

The author's copy-theory of movement, according to which a trace is a copy of the moved element which is deleted in the phonological component, but remains available for interpretation at LF. Under this view, the operation Move is a complex operation comprised of (at least) three suboperations: (1) a suboperation of deletion (2) a suboperation of merger; and (3) a suboperation of trace deletion. In addition, Move should be followed by an operation of chain formation relating the relevant copies. There are several conceptual inadequacies in this picture. First, if no explanation for why lower copies must be deleted in the phonological component is provided, the notion of a trace as a primitive is reintroduced. To put it more generally, the simplest and, therefore, most desirable version of the copy theory of movement should take traces and heads of chains to be subject to the same treatment as heads of chains and traces, such as phonetic realization, for instance, should follow from independently motivated properties of the computational system, rather than being idiosyncratic properties of the chain links themselves.

Deletion of traces (lower chain links) becomes even more enigmatic, if we adopt the core Minimalist assumption that economy considerations play a role in determining the set of admissible derivations in a given language or universally. Consider for instance the structure in (1) below, where John has moved to the subject position and left a copy behind. The derivation of (2a) from (1) requires one application of deletion targeting the lower copy of John, apparently being less economical than the derivation of (2b), which involves no application of deletion. Thus, if the derivations of (2a) and (2b) were to be compared in terms of economy, we would wrongly predict that the derivation of (2b) should rule out the derivation of (2a).

(1) [John [eat [arrested] John]]

(2) a. John was arrested.
   b. * John was arrested.

Another conceptual problem with the computational system as proposed in Chomsky (1994, 1995: chap. 4) is that Merge is taken to be an operation in its own right in certain cases, and a suboperation of other operations in other cases. In an optimal system, we should in principle expect Merge to have the same theoretical status in every computation. Finally, as is emphasized by Brody (1996), if chain formation and Move express the same type of syntactic operation, the theory which contains both notions is redundant.

This dissertation develops a strictly Minimalist version of the copy theory of movement which overcomes the conceptual problems raised above and has a broader empirical coverage than the versions developed in Chomsky (1993, 1994, 1995: chap. 4). It proposes that the fact that a chain cannot live more than one link evenly realized (see (2a)) follows from Kayne's (1994) Linear Correspondence (LCA), according to which the linear order of a PF sequence is determined by asymmetrical c-command. Under the assumption that the two copies of John in (1) are "nonidentical" (they relate to the same element in the initial phrase), no linear order can be established in accordance with the LCA. Given that the verb was in (1), for instance, asymmetrically c-commands and is asymmetrically c-commanded by the "same" element, namely John, the LCA should require that not precisely John, but John itself, c-commanded the antecedent of John, violating the asymmetry condition on linear order. Put simply, deletion of all but one link is furred on a given chain CH in order for the structure containing CH to be linearized in accordance with the LCA. The derivations of (2a) and (2b) therefore cannot be compared for economy purposes, because only the former yields a PF object.

The next question to be addressed then is why it is the case that only traces are deleted for purposes of linearization, but not heads of chains.
Chapter II ("Theoretical Framework") devotes 176 pages to laying out and critically analyzing the essentials of minimalist syntax, adjusting whatever the latter constructive chapters demand. For instance, one of the technically more sophisticated maneuvers in the elaboration of Predicative Syntax. Likewise, in this chapter, Nunes shows how we can dispense with the Uniformity Condition on chains, which Chomsky (1985) employed in capturing the Structure Preservedness within his Bore Phrase Structure model. What looks somewhat awkward is Nunes's proposal to put strong features, which are, after all, the driving force of overt movement operations, into the set of phonological features of the lexical items bearing them. Given that strong features have to be checked in syntax, this conflicts with the otherwise fairly plausible (and restrictive) assumption (1).

(1) (Overt operations cannot detect phonological features at all.)

Chomsky (1985a, p.330)

Copy + Merge Theory of Movement

Let's now turn to the core issue of Nunes's dissertation, the "Copy + Merge Theory of Movement." As is well known, minimalist syntax rules on two-structure-building operations, namely, Merge and Move. Merge transforms two (unseriated) trees T1 and T2 into a single (unseriated) tree T3 composed of T1 and T2, as exemplified in (2). Take all sentences to be unseriated, unless stated otherwise. I use traditional notation in order to cut out issues tangential to this review.

(2) Merge: T1 \text{ (arrest)} \rightarrow T2 = \{\text{John} \} \rightarrow T3 = \{\text{John, arrest}\}

Several conditions can be— and usually are—imposed on Merge. Thus, T1 and T2 can be required (proper) subparts of T3 in the technical sense. This is usually formulated as a cyclicity condition or ban on 'internal Merge'. Secondly, the operation Move, later called 'Attract' (Chomsky 1996a, chapter 4), transforms a single tree T1 into a tree T2 in the way specified by the following suboperations:

(ii) locate two subtrees T3 and T4 of T1, T3, the attractor, c-commanding T4, the attracted element;
(iii) verify that T3 and T4 contain features able to establish a c-relationship; (T4 is the Last Resort requirement. It could also be part of (i), assuming that T3 only attracts the right kind of object.)
(iv) verify that there is no subtree T5 of T1 closer to T3 than T4, such that T3 and T5 could establish a c-relationship equivalent to the one specified in (ii);
(v) copy T4;
(vi) form a chain between T4 and T4;
(vii) check the features in an established c-relationship between T3 and T4 (and T47); The ordering of (v) and (vi) as well as the uncertain status of T4 with respect to (vii) raise non-trivial questions.
(viii) Mark T4 as phonetically unrealizable. (This step is trivial in the covert component.)

A much simplified example of Move is given in (3).

(3) (\ldots \text{ encode the PF condition on T4} \ldots)

Copy + Merge Theory of Movement

Let's now turn to the core issue of Nunes's dissertation, the "Copy + Merge Theory of Movement." As is well known, minimalist syntax rules on two-structure-building operations, namely, Merge and Move. Merge transforms two (unseriated) trees T1 and T2 into a single (unseriated) tree T3 composed of T1 and T2, as exemplified in (2). Take all sentences to be unseriated, unless stated otherwise. I use traditional notation in order to cut out issues tangential to this review.

(2) Merge: T1 \text{ (arrest)} \rightarrow T2 = \{\text{John} \} \rightarrow T3 = \{\text{John, arrest}\}

Several conditions can be— and usually are—imposed on Merge. Thus, T1 and T2 can be required (proper) subparts of T3 in the technical sense. This is usually formulated as a cyclicity condition or ban on 'internal Merge'. Secondly, the operation Move, later called 'Attract' (Chomsky 1996a, chapter 4), transforms a single tree T1 into a tree T2 in the way specified by the following suboperations:

(ii) locate two subtrees T3 and T4 of T1, T3, the attractor, c-commanding T4, the attracted element;
(iii) verify that T3 and T4 contain features able to establish a c-relationship; (T4 is the Last Resort requirement. It could also be part of (i), assuming that T3 only attracts the right kind of object.)
(iv) verify that there is no subtree T5 of T1 closer to T3 than T4, such that T3 and T5 could establish a c-relationship equivalent to the one specified in (ii);
(v) copy T4;
(vi) form a chain between T4 and T4;
(vii) check the features in an established c-relationship between T3 and T4 (and T47); The ordering of (v) and (vi) as well as the uncertain status of T4 with respect to (vii) raise non-trivial questions.
(viii) Mark T4 as phonetically unrealizable. (This step is trivial in the covert component.)

A much simplified example of Move is given in (3).

(3) (\ldots \text{ encode the PF condition on T4} \ldots)
rate and expanded operation. (There are various options as to where exactly (vii) should go, which I sidestep.) Crucially, (viii) is nothing to worry about at all outside the PF-branch of the computation. Even the LCA and the GLP-branch will all be phonetically realizable.

The question that comes up is how such a system gets rid of the superfluous copy/copy at PF? The answer, which makes up the bulk of Chapter 2, is that pairs of Chains at PF (or Why Traces are Not Phonetically Realized?) has to do with Kayne’s (1984) Linear Correspondence Axiom (LCA). The LCA roughly states that asymptotic c-command between non-terminal nodes is determined by the precedence among the terminals dominated by the respective non-terminals. This principle is adapted by Nunes in the following way.

(4) Linear Correspondence Axiom (LCA)
A category y precedes a category x if:
(i) y asymmetrically commands x or
(ii) y precedes x and dominates x (p. 177)

For the sake of clarity, we take categories to correspond to subtrees. At PF, the complex operation Linear Correspondence realism trees in line-ordered ones on the basis of (4). Precedence among subtrees as it emerges from (4) is isomorphic to the familiar precedence relation among nodes of binary trees (cf. Partee et al. 1963, p. 441), although exceptions arise where the correspondence between categories and subtrees breaks down and where c-command is undefined. Like precedence among nodes, precedence among subtrees is required to be a strict, partial order (transitive, irreflexive, and asymmetric). Crucially, this property is taken to be violated by structures that contain multiple copies of the same subtree, such as (5).

(5) \[ \text{f --> John, was > John, arrested John.} \]

Precedence among subtrees will contain both <John, was > and <was, John>, because the occurrence of John in SpecIP asymmetrically c-command was and was asymmetrically c-command the occurrence of John in VP (275). Thus, precedence among subtrees, although not asymmetric, is not well-formed and Linearize cannot proceed, that is, the derivation does not converge. Note that to derive such a result, asymmetric c-command must be able to distinguish occurrences of John, while precedence among subtrees isn’t. Slightly paradoxically, <John, John> must be a member of asymptotic c-command in one processing, whereas among subtrees will be neither asymmetric nor irreflexive (p. 273). These subtle distinctions, of course, have to be based on clear criteria of identity together with an exact specification of the stratum with respect to which ordering properties are verified. The former requirement is meant to be met by the following assumption.

(Take a term of a given structure to be inherently distinct from all the other terms of that structure unless it is specified not to be a copy. In other words, the output of a copying operation cannot copy a term that was interpreted at the C1 interface as non-distinct from T or other copies of T and distinct from every other term by default. The copying operation must be the ‘same’ as the original term, i.e., if T has no index, Copy targets T, assigns an index i to it and creates a copy of the indexed term; or, if T was already indexed, a copy must be created such that it creates a copy of the indexed term.” (p. 86)

As with categories before, we can take terms to correspond to subtrees unless stated otherwise. Crucially, coindexation makes subtrees nondistinct, a property geared to the purpose of treating two objects as one and the same with respect to certain operations. Reference to the C1 interface is at least suggestive of that. Analogously, Linearize is close enough to the A-P interface to call it an “interpretable” operation, so construction of precedence among subtrees as part of Linearize can be considered sensitive to nondistinctness, while verification of asymmetric c-command isn’t.

Thus far it looks as if copying before Spell-Out inevitably leads to non-wellformedness at PF, an absurd consequence if movement is to be captured by Copy + Merge. But before we see how Nunes moves this observation into a virtue, let’s have a closer look at Linearize.

Clearly, precedence among subtrees is not a linear order in the technical sense, that is, it is a partial not a total order. Precedence according to (4) will only be partially ordered vertically, i.e., by dominance. Only precedence among terminals can be required to be a linear order, derivable from the locally linear order induced on non-terminals by asymmetric c-command (Kayne 1994b). Nunes is not always terminologically careful enough to mark these differences. Yet, apart from open issues related to the absence of a traditional distinction between terminals and non-terminals and the ban on non-branching projection in minimalist bare phrase structure (for discussion see section II.17 and Chemisky 1995b), there seems to be good reason for Nunes to have Linearize construct precedence among subtrees and not just precedence among terminals. Consider example (6).

(6) \[ \text{f --> John, was > John, arrested John.} \]

Coping has applied to the complex DP, so that where the index went. If we take X-heads as terminals, then precedence among terminals of (6) is induced by asymmetric c-command by definition (4) would contain pairs such as <boy, was> and <was, boy>. Without indices it is unclear how to treat the two occurrences of boy. They are definitely not identical and the pair can’t be derived according to which “fx” and “fy” are identical, then there is no rule of x (Smolyn 1971, p. 35). One occurrence is a part of the specifier of IP while the other is part of the complement of V. Does this mean there is a difference? As already mentioned earlier in the case of occurrences of subtrees with identical structural content? That would depend on the exact implementation of such a default. (It is doubtful, by the way, whether anything but another layer of indirection is needed to get the well-formed representations of bare phrase structure.) Thus, unless indices percolate to each subtree of a subtree copied, precedence among subtrees is the proper place to rule out (6); as it stands, obvious precedence among subtrees will contain <(IP, boy), was> and <was, (IP, boy)>. Coindexation means nondistinctness, and the latter leads to a violation of the asymmetry and irreflexivity requirement put in place by the Linearize cannot proceed and (6) is ruled out as a pair. (Just after finishing this review, Nunes informed me (p.e.) that it is index percolation he would favor in dealing with cases like (6). So, things are changing all the time.)

Let’s postpone inspecting the solution to the already mentioned apparent impasse created by (5) and (6) for another day and consider a conceptually interesting issue first.

LCA and the Minimalist program
Kayne’s LCA is a powerful tool that almost doesn’t require anything about the language structures of natural language sentences should have. It is doubtful, whether the highly derivational Chomskyan system, by and large adopted by Nunes, is able to take full power of the LCA, if the sole aim is to achieve PF-linearization. Note first that Kayne does not seem to take copies of movement to lead to LCA-violations (Kayne 1986, p. 135, fn. 3). But, how else could reductive minimalist pairs emerge among terminals or precedence among terminals? The former case clearly requires that some subtree asymmetrically c-command itself. Such a thing could result from relaxing the “Single Model Condition” on dominance (N = node, ID = immediately dominates; cf. Blevins 1990, p. 48).

A subtree T1 allowed to be immediately dominated by two distinct nodes x and y, such that either x dominates y or y dominates x, should yield <T1, T2> as a member of asymmetric c-command. Hence, in the case of Merge twice in the Chomsky system, that is, it cannot acquire two mother nodes, such a violation simply cannot arise. The irreducibility condition, thus, holds trivially. Next, violations of asymmetry would either follow from the same kind of relaxation of (7), namely, if there is a distinct node z dominance-wise between x and y, such that z immediately dominates an additional subtree T2. T2 could then both asymmetrically c-command and be asymmetrically dominated by y. But, this is a non-issue as we’ve just seen. Alternatively, two subtrees could symmetrically c-command each other. Crucially, the latter case has been obviated in the minimalist system by stipulation. Nunes’s version of this hinges on the inaccessibility of intermediate projections and lower segments of adjunction structures.

(8) C-Command
Where e and d are accessible to the computational system, e c-command d if:
(i) e does not dominate d
(ii) a \[ \text{a} \]
(iii) every category dominating d dominates (p. 177)

Sidestep further complications, we can say that whenever subtree T1 adjoins to subtree T2, the resulting sister of T1 will not be accessible any longer. T1 can asymmetrically c-command into its sister and therefore precede every subtree dominated by its sister. Likewise, whenever two complex subtrees undergo ‘non-adjectival’ Merge, the one that projects won’t be accessible any longer, so the non-projecting one can asymmetrically c-command into its sister and precedence among the respective sets of subtrees is fixed. Thus, except for the case of two terminal subtrees in a sisterhood relation — for which an extraneous assumption is required because, as indicated above, nonbranching projection isn’t allowed — no violation of asymmetry can arise. Again, compliance with that constraint has been built into the system and need not be invoked as an extraneous condition on representations.

We could, for the sake of argument, allow the LCA to do some genuine work in filtering out specifiers in the standard sense (i.e. sisters of intermediate position), but that seems like an ad-hoc solution to the problem of ‘specifiers’ along the lines proposed by Kayne (1994). Intermediate projections would then have to be accessible with respect to c-command as defined in (8). Structures like (9) would then give rise to an asymmetric violation.

(9) f --> f, the cat (1 is in the main line)

Given that among other things X asymmetrically c-commands cat and that YP asymmetrically c-commands is, both cat, is greater and cat, should hold for purposes of c-command, an apparent violation of precedence among terminals. This violates the LCA. It is unclear, however, how to represent and exclude this in Nunes’s system. Once again, indices, being reserved for copies, would apply here. Thus, the case in which asymmetric (identical) subtrees (i.e. subtrees to which Lehnin’s principle does apply) get into conflict with the asymmetry condition on linearization doesn’t receive as straightforward a treatment as one would expect. (An attempt to derive banning on any branching structures, for a z, would be defective in the same way.) But if the canonical cases the representational LCA condition was meant to capture are not an issue in a theory invoking the LCA, this is likely to cast some doubt on the LCA based approach to the problem caused by stipulated quasi-identical (“nondistinctives”) subtrees, i.e. copies/ traces, really is. There surely is a very creative idea behind all of that, but its implementation doesn’t come for free.
In fact, minimalist systems could employ some very simple top-down algorithm for converting unordered trees into ordered trees. Hitting on a pair of sister-subtrees T1 and T2, plus and minus accessible respectively, let T1 precede T2. Hitting on a pair of sister-subtrees, both accessible, let the one that is an X'-head precede the other subtree. This is all there has to be done (modulo the already mentioned extra assumptions concerning X'-sisters at the bottom of the tree).

Most importantly, never does the system have to keep track of the content of any subtrees through-out linearization. This would be necessary if one were expecting one and the same subtree to (illicitly) appear twice in the resulting ordered tree, an impossibility as far as the ones that obey Leibniz's principle are concerned. For Nunes's solution to work, however, there has to be an additional check of the resulting ordered tree in order to detect nondistinct subtrees. This would seem to amount to pair-wise inspecting a full chart of precedence among subtrees (or precedence among terminals), on the size of which, as is well-known, there is no upper bound.

I have been pushing this rather abstract point at the risk of the author doesn't fully subscribe to. It is my strong impression, though, that the delicate issue of syntactic individuals and how they are structurally individuated affects Nunes's approach in a potentially adverse fashion, even if the determinate algorithms make this look more or less transparent. Let's now return to the question why (5) ultimately gets realized as (10) at PF.

This is due to the possibility of Chain Reduction operating in the PF-component before Linearize applies.

(11) Chain Reduction

Delete the minimal number of terms of a nontrivial chain CH which suffices for CH to be mapped into a linear order in accordance with the LCA. (p. 315)

Suppose the two occurrences of John, have formed a chain in (5). Since each copy consists of one subtree, deleting any one of the two will allow compliance with the LCA. Asymmetrical command as computed over the thus reduced tree structures will not produce any offensive pairs involving nondistinct copies. Therefore, precedence among subtrees will still be preserved in a partial order structure. Alongside of (10), (12) would be another well-formed output of Linearize.

(12) was arrested John

Even the option of deleting both occurrences of John, leads to an LCA compatible structure. However, Nunes assumes that transderivational economy blocks the application of two deletions where one such operation would likewise have produced a convergent output. Thus, (12) is an impossible representation of (5).

(13) was arrested

In sum, Nunes gives the following answer to the first of the fundamental questions he raised. Phonetically realizing more than one member of a nontrival chain leads into conflict with the LCA. Phonetically realizing no member of a nontrival chain is blocked by transderivational economy, which prevents Chain Reduction from applying more than once. The survival of one member is compatible with linearization to proceed, so deleting that member is a superfluous step.

On a more conceptual note, it must be said that invoking Chain Reduction instead of just "Reduction" pays tribute to the fact that, even if Move is dismantled as a unified operation, the effects statistically attributed to it have to be dealt with in a special way, at least to some extent.

This problem is dealt with by the assumption that [when participating in an overt checking relation, a [-interpretable] feature can optionally be checked with respect to PF] (p. 307)

Visibility means that FF-elimination doesn't have to deal with such a feature, that is, the X'-feature of John, and the DP-feature of who, in SpecCP of (17b). Optionality of checking [-interpretable] features with respect to PF is required to properly handle successive cyclic movement (17c). Whether the intermediate SpecCP were to check its D-phrase, the final step into the matrix SpecCP would, in the best case, produce a copy that contains equally few PF-relevant features as the one in the intermediate position. Thus, (10) should be an optional realization of (17c).

(14) You asked us to be arrested John. Or: Did they arrest who?

(15) a. was arrested

Assuming that Linearize ultimately produces a string of X'-heads (p. 274), there is still another step to take before a phonetic representation x can be constructed. From the perspective of prosodic phonology it is far from clear that a string of terminalis is sufficient. Rather, it would seem, that mapping into prosodic constituents has to start from ordered syntactic constituents, so the strategy of constraining and evaluating precedence among subtrees rather than precedence among terminals as discussed above would be vindicated on those grounds. The X'-elements still contain formal features. These will be eliminated on an item-by-item basis in accordance with (16).

(16) Formal Feature Elimination (FF-elimination)

Given a sequence of X's (\( X' \times X' \times X' \)), where each X' is either an input to the next X or an output from the previous X, the X that is the input of Linearize, P and F are all formal features and P is a set of phonological features, delete the minimal number of features such that each set of formal features in order for P to satisfy Full Interpretation at PF (p. 315).

Crucially, (15a) and (15b) differ in that the sub-tree John in the former already lacks its Case-feature, due to having checked it off in the checking domain of 1st. This saves one application of FF-elimination. The sub-tree John in VP of (16b), however, does have a Case-feature still unchecked. Thus, FF-elimination will have to apply at least one time more. Other things being equal, derivational economy therefore chooses (16a) and blocks (16b). It is quite easy to see now why the hierarchy of the most prominent member of a chain reaching the PF-branch gets realized. Movement is driven by feature checking. The functional projections dominate the lexical ones. And, there is no lowering. The three instances of deletion, Checking, Chain Reduction, and FF-elimination will interact in the desired fashion if it is exactly the hierarchically most prominent copy reaching PF that has fewer features due to checking. Chain Reduction targeting entire subtrees doesn't care about the number of features present in the chain members deleted. But PF-elimination makes a difference. Thus, the more features can be gotten rid of via Chain Reduction the fewer times FF-elimination has to apply. This logic favors the survival of the copy that has undergone a checking operation earlier, i.e., the hierarchically most prominent one.

Further adjustments are nevertheless required since the ECM command (17a) and X'-movement (17b), as well as the general question of successive-cyclical movement (17c).

(17) a. a. was arrested John.

b. [\( \text{who} \), who, who, did you think \( \text{who} \), who, arrest \( \text{who} \), who, who, who]

c. [\( \text{who} \), who, who, who, who, did you think \( \text{who} \), who, who, who, who, who, arrest \( \text{who} \), who, who, who, who, who]

If movement of John, in (17a) and who, in (17b) only checks interpretable features, the two non-distinct copies still carry the same number of formal features at PF. FF-elimination should optionally allow (16a) and (16b) to be well-formed.

(18) a. We expected to be arrested John

b. Did they arrest who?

Assuming the 'primary' copy to e-command the 'parasitic' one, these two cannot form a chain, because that chain doesn't meet the Last Resort requirement. Neither of the two enters any checking relation. At the same time, Form Chain between the copier in operator position and the 'parasitic' one violates the MLC, given that the 'primary' copy intervenes. But, if the 'parasitic' copy cannot enter any chain, the familiar LCA violation results. Again no extra assumptions are necessary to derive this. Structures like (24) reveal another sublety of Chain Reduction. (24b) could be saved by deleting both copies of the 'primary' chain, which would yield PP-output (25b).

(25b) Did you file without reading which article

This is prevented by conditioning Reduction (11) on linearizability of just the chain being reduced, irrespective of whether the entire structure will eventually satisfy the LCA. Thus, the linearization algorithm has to be even more fine-grained in being able to keep track of which chain substructures belong to.

Two further issues come to mind immediately in connection with Norns treatment of parasitic gaps that need to be explored. A theory heavily relying on e-command as required by the LCA must also be very explicit about the exact structural positions of extraposed constituents. If, for example, the difference between (22) and (24a) is to be captured. Rightward adjunction not being an option, the author proposes a coordination-like treatment of adjuncts, the full detail of which would have to be worked out. Secondly, one has to worry about cases of parasitic gaps for which the Anti-Command condition doesn't seem to hold.

This issue has recently been brought up again by Brody (1995) in connection with Hungarian examples like (28).

(28) Róka, atemberaubend!; ha t, a játszók és a hegedüslők velünk
Who would you like if I came without you having invited me?
(Brody 1995, p.94)

Remnant movement

Let me finally discuss the phenomenon of Remnant Movement which leads to 'sideways movement' of a more complex kind, as (27) shows (cf. Müller 1997).

(27) a. ...elected by the committee John never was
b. I abortion (Johns), by the committee (Johns), never was I abortion (Johns), by the committee ()

Chains can be formed between John, in SpecIP and John, inside the VP in situ, as well as between the frontal VP and the VP in situ. Copying VP, however, has produced another copy of John, inside the frontal VP. For lack of e-command, that occurrence of John cannot undergo Form Chain and thus won't be deletable under any kind of Chain Reduction. Consequently, (23a) cannot be derived in the system as it stands.

In sum

Summing up, it must be emphasized again that in spite of the various weaknesses pointed out, this dissertation covers an admirably broad range of syntactic issues, creating deductively nontrivial links between seemingly disparate phenomena.

References


Stabler, Edward (1996) Acquiring and Parsing Languages with UCLA.


The clicks behaving 'just' like other consonants.

Most of the tracks do not lend themselves easily to analysis, but they all have something interesting to tell us. For instance, Track 3 has no accompanying transcription or even translation but its use in class may have a great impact on students. This recording, and an unpublished manuscript of notes by Robert Stokey, are all we know about the Khoisan language. When students realize there are still many un- and under-described languages (including many Khoisan languages) in the world, it underscores part of the adventure of linguistics. The recording presents Khoisan as an anomaly for classification, by having both bilabial clicks and uvelar stops (characteristics of non-Khoisan, Southern Khoisan languages) and the presence of a word that looks cognate with Khoi languages.

All in all, the CD is a great deal of fun and should be a part of any Linguistics Department's collection. The only distraction that I can see is the high cost of the CD but one should keep in mind that the CD is being offered at cost.

LEFT WITH THE SOUNDS

by Bonny Sands

reviewing the CD

Extinct South African Khoisan Languages

"Know me through my tongue", said Muklap, one of the last fluent speakers of Ora, in 1938, to an international meeting of phoneticians. The recording featuring his words, spoken in Ora are especially poignant. If linguists forget about a language, who else will remember it?

Prof. Tony Trill of the University of the Witwatersrand had dug through the archives to find recorded examples of some of the "forgotten" languages of South Africa. As the leading expert on these languages, he translated and transcribed where it was possible, and resurrected voices from old, scratched wax cylinders.

A booklet accompanies the CD, and includes transcriptions, translations, pictures, and introductions to each of the 15 tracks. The pictures are as moving as the words. For instance, we see pictures of [Khaku and his family from 1938, and also pictures of two women who heard Khomani as children, listening to [Khaku's words.

Ora, [Xegwi, Ku [Xhosa, [Xhosa are now extinct, but thanks to this CD, we can hear them for the first time.

Track 2 is a good recording to use in introductory linguistics classes. With its IPA transcription, students can try to follow along with "The North Wind and the Sun" in [Xegwi. It is fun for students to hear a click language as it is spoken, with the clicks behaving 'just' like other consonants.

Most of the tracks do not lend themselves easily to analysis, but they all have something interesting to tell us. For instance, Track 3 has no accompanying transcription or even translation but its use in class may have a great impact on students. This recording, and an unpublished manuscript of notes by Robert Stokey, are all we know about the Khoisan language. When students realize there are still many un- and under-described languages (including many Khoisan languages) in the world, it underscores part of the adventure of linguistics. The recording presents Khoisan as an anomaly for classification, by having both bilabial clicks and uvelar stops (characteristics of non-Khoisan, Southern Khoisan languages) and the presence of a word that looks cognate with Khoi languages.

All in all, the CD is a great deal of fun and should be a part of any Linguistics Department's collection. The only distraction that I can see is the high cost of the CD but one should keep in mind that the CD is being offered at cost.