Evidence for Survive from covert movement

Winfried Lechner
University of Athens

The paper pursues two goals. First, it motivates a particular view of the Survive principle. Concretely, it is suggested to interpret the Survive principle as the syntactic instance of a more general push-up mechanism that is responsible for triggering movement induced by type incompatibility on the semantic side. Second, I identify a particular set of properties that the Survive analysis predicts for configurations involving multiple covert movements. These diagnostics, which help to discriminate between survive and Attract based models of dislocation, are argued to be manifest in scope restrictions on double object constructions and inverse linking. The critical factor setting apart the two models consists in the observation that only the Survive principle is able to express ordering restrictions between different types of movements (Case driven movement vs. QR) in a natural way. The resulting analysis also supports the phonological theory of QR.

1. Introduction

Traditional historical phonology distinguishes between *drag chains* and *push chains*. In drag chains, diachronic change is triggered by a gap in the system which is filled by a newly emerging element (phoneme). If the gap is thought of as an external *attractor*, the fundamental properties of this model closely resemble an Attract based theory of syntactic dislocation, in which a higher attracting head induces movement of a lower category (Chomsky 1995, 2005, a.o.). By contrast, changes which are associated with push chains are initiated by the emergence of new elements that expel parts of the system from their original position. Conceived this way, push chains create effects similar to those which *The Survive Principle* (Stroik 1999, 2009) postulates for syntactic dependencies.[1] Or, put the other way round, the Survive Principle imports a concept similar to push chains into syntax.

In essence, the Survive Principle (henceforth TSP) formulates an alternative for motivating syntactic dislocation. Unlike Attract based models, which make movement contingent upon properties of higher heads, TSP locates the trigger for dislocation exclusively in the relation between the moved category and its local environment.

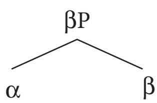
1. On push chains in syntax see also Preminger (2008); Putnam (2007); van Riemsdijk (1997) and Stroik (1999, 2009), among others.

Concretely, TSP states that if a node α is not feature compatible[2] with its sister node, α is pushed into a higher position in order to 'survive' the consequences of feature mismatch. Moreover, the target of movement is determined by the next new head which is introduced into the derivation. One way to make the TSP explicit is as in (1):

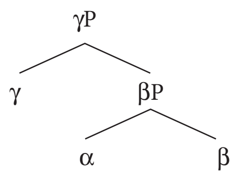
(1) *The Survive Principle (first version)*
For any nodes α, β and γ:
If α is not feature compatible with its sister node β,
(i) merge the projection of β with a new head γ and
(ii) remerge α with a projection of γ.

(2) schematically illustrates how the TSP triggers dislocation of an α which is not feature-compatible with its sister node β (2a)[3]. In an initial step of the derivation, a new head γ is externally merged with a projection of β (2b). As a consequence of TSP, the incompatible category α is expelled from its original position, moving to a projection of the newly merged head γ (2c):

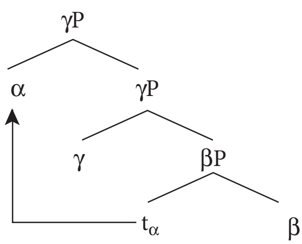
(2) a. α feature-incompatible with β



b. Merge new head γ



c. Move α to a projection of γ



2. Roughly, two nodes are feature incompatible if their original feature matrices prior to any checking operation have an empty intersection. I will not attempt to make the notion of feature incompatibility(more precise here. Stroik(s original version of TSP is (i):

(i) The SURVIVE Principle (Stroik, 2009)
If Y is a SO [Syntactic Object] in an XP headed by X and Y has an unchecked feature incompatible with (i.e., cannot potentially be checked by) the features of X, Y must Remerge from the WorkBench with [a projection of; WL] the next head Z that c-commands XP.

3. For typographic convenience, complements are drawn on the left.

The present paper explores a possible venue for locating evidence in support of TSP based on interpretive properties that are also syntactically encoded. In particular, I will propose that certain restrictions on relative quantifier scope in English can be given a simple explanation if TSP is adopted, but require additional assumptions, which essentially mimic the effects of TSP, in the standard Attract based model.

The analyses to be presented cover two different phenomena: scope freezing in the double object construction (Barss & Lasnik 1986; Richards 2001a; Bruening 2001; Sauerland 2000), and scope restrictions in contexts involving Inverse Linking (Larson 1987; Heim & Kratzer 1998; Sauerland 2000). As a common thread, both analyses will be seen to rest on the assumption of intermediate traces. Such traces are automatically generated by the TSP model, but not by theories that use Attract as a trigger for movement. This finding provides support for integrating the concept of TSP into the syntactic component.

The paper is structured as follows. In Section 2, I discuss relevant background assumptions pertaining to the specific implementation of TSP that will be used (2.1) and the particular model of the grammar (2.2). Section 3 introduces two scope restrictions and develops a common TSP analysis of these phenomena. Section 4 summarizes and comments on the results.

2. Background assumptions

2.1 The survive principle and type driven interpretation

The principle of *type driven interpretation* (TDI; see (3)), to be adopted here, expresses the widely endorsed view that certain covert movements are induced by the need to repair type incompatibilities:[4]

(3) *Type driven interpretation*
If a category α is not type compatible with its sister node β, move α to the next higher position type compatible with α.

4. For discussion see e.g., Heim and Kratzer (1998). α is type compatible with β if either (i), (ii) or (iii) applies:

(i) $[\![\alpha]\!] \in D_{\langle\tau,\sigma\rangle}$ and $[\![\beta]\!] \in D_{\langle\tau\rangle}$ (Function Application, with β as argument)
(ii) $[\![\beta]\!] \in D_{\langle\tau,\sigma\rangle}$ and $[\![\alpha]\!] \in D_{\langle\tau\rangle}$ (Function Application, with α as argument)
(iii) $[\![\alpha]\!] \in D_{\langle e,t\rangle}$ and $[\![\beta]\!] \in D_{\langle e,t\rangle}$ (Predicate Modification)

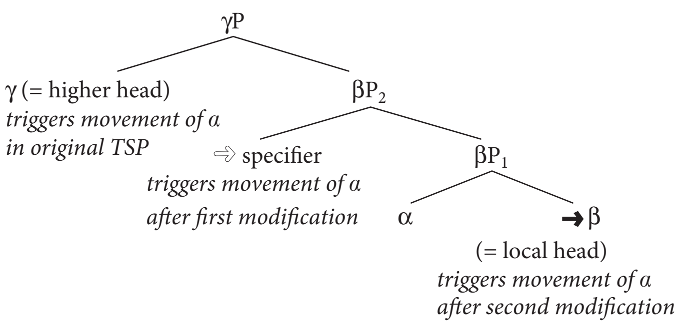
Interestingly, the formulation of TDI, which triggers covert dislocation, bears an uncanny resemblance to TSP in (1). To bring out the similarities more transparently, (1) can be rephrased as in (4) below:

(4) *The Survive Principle (first version, paraphrased)*
If a category α is not feature compatible with its sister node β, move α to the next higher position projected by a new head γ.

In particular, both principles include an antecedent clause defining incompatibility (type or feature wise) and specify a movement strategy to resolve this conflict in the consequent of the conditional. But while the antecedent clauses of (3) and (4) are identical *modulo* the distinction 'type' vs. 'feature', the consequents are given a different wording in each principle. For TSP, the escape strategy consists in movement to a position above the next head, while for TDI, movement must target a node of suitable type.

In what follows, I argue for three changes in the definition of TSP that will render this relation more symmetric and at the same time contributes to an overall simplification of the Survive model. The first change will make TSP look more like TDI, thereby removing an imbalance in these two otherwise quite similar principles. The second modification eliminates a stipulation from TSP, rendering it more general than the original version. These two changes affect TSP *incrementally* in that they expand (4), which creates intermediate landing sites only upon insertion of new heads (represented by γ in (5)) to systems that add specifiers (see ⇨ in (5)) and local heads (see ➜ in (5)), respectively, to the list of triggers for movement:

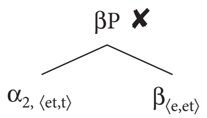
(5)



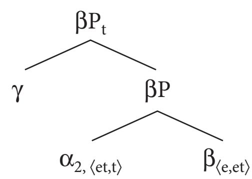
Finally, a last revision extends the range of categories affected by TSP by generalizing the definition of 'compatibility' to include type mismatches, in addition to feature incompatibility.

Turning to the first modification, the contrast between TSP and TDI observed above is a function of another, deeper difference between the two principles. While TDI assumes that heads combine with their complements in the same way in which specifiers combine with heads (or, more precisely, with nodes containing heads and their complements), TSP ignores specifiers all together. To illustrate the relevance of specifiers for TDI, consider the text book case of type-driven movement of an object quantifier in (6).

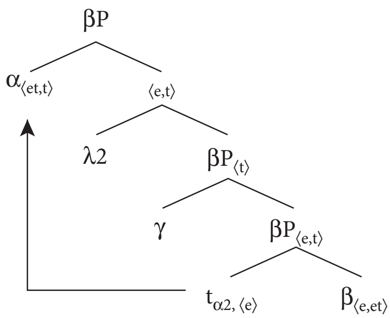
(6) a. α type incompatible with β



b. Merge specifier of β (=Y)



c. Move α to projection of β



In (6a), the generalized quantifier type expression α_2 (type ⟨et,t⟩), originates as a sister node to a transitive verb (= β). On standard assumptions, α needs to attach next to a node of type t in order to be able to combine with the rest of the clause in semantics. Such a node is provided by addition of the *specifier* of β (= γ), i.e., the subject, in (6b).[5] Thus, specifiers are instrumental for TDI in that they create suitable landing site for QR (see (6c)).

I would like to suggest to generalize TSP in the same direction, such that TSP driven movement is not only triggered by new heads, but also by the addition of specifiers. One way to achieve this is made explicit by the revised version of the TSP in (7):

(7) *The Survive Principle (second version)*
For any nodes α, β and γ:
If α is not feature compatible with its sister node β,
(i) merge the projection of β with a new category γ, resulting in δ, and
(ii) remerge α with δ.

5. I assume, as is standard (implicit) practice, that phases and quantifiers make their semantic type visible to syntax.

(7) derives the desired effect of pushing α across newly merged heads as well as across newly added specifiers by not restricting the value of γ to heads. To see how this works, assume again that α is in complement position of a verb. If the projection of β (= VP) merges with a head γ (e.g., v^0), then α will attach to a projection of v^0. Subsequently merging VP with a specifier (e.g., a subject) results in movement of α to a position adjoined to the minimal node containing the specifier and vP, as illustrated by (6c).

The second change of the TSP pertains to the role of heads and eliminates an extrinsic statement from the Survive model. In its current incarnation (7), TSP demands that α moves to the next higher position projected by a new head γ. However, confining movement to contexts in which a *new* head – i.e., a category that is not yet part of the derivation – is inserted only adds a stipulation to TSP and is, as far as I can see, not motivated by empirical considerations. I will therefore assume that movement is triggered after any application of merge, irrespective of whether the externally merged category is a new head or the local head that α has been first merged with, or a specifier. This can be informally summarized as in (8), which provides the pre-final version of TSP:[6]

(8) *Survive Principle (pre-final version)*
For any nodes α, β and γ, such that γ is the mother of α:
If α is not feature compatible with β, then
(i) adjoin α to the result of merging α with β, if β is a head and
(ii) adjoin α to the result of merging γ with β otherwise.

The definition (8), which essentially mandates that any feature-incompatible α move to the next branching node in the tree, includes two subcases. If α serves as the complement of a head, say V^0, then α will be pushed to a VP-adjoined position by (8)(i). If, on the other hand, α is adjoined to a maximal projection, say vP, which is still to combine with a specifier, e.g., the subject, then α will land in an outer adjunct to vP by (8)(ii).

The last modification of TSP removes another unnatural stipulation from the definition of Survive, resulting in further reduction of redundancy. In particular, I would like to suggest that 'compatibility' is not restricted to 'feature

6. An alternative and somewhat more natural definition is provided by (i):

(i) For any nodes α, β and γ:
If α is not feature compatible with its sister β and (i), (ii) and (iii) holds, then adjoin α to the result of merging α with γ.
(i) γ reflexively dominates α and
(ii) γ is merged with β and
(iii) β does not contain α

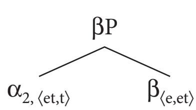
compatibility', but extended to also comprise *type* incompatibility. The final definition of TSP is given in (9):

(9) *Survive Principle (final version)*
For any nodes α, β and γ, such that γ is the mother of α:
If α is not feature or type compatible with β, then
(i) adjoin α to the result of merging α with β, if β is a head and
(ii) adjoin α to the result of merging γ with β otherwise.

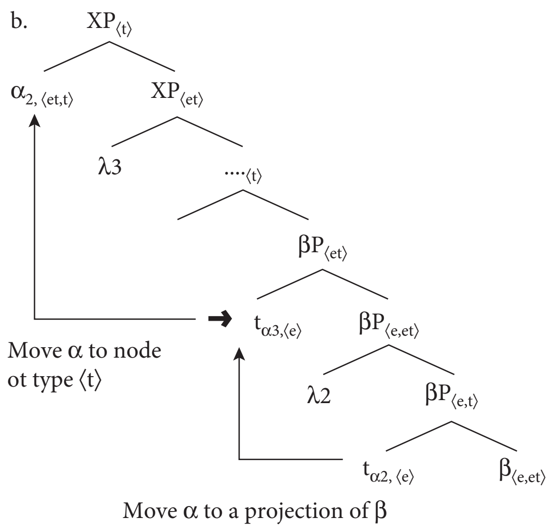
(9) can be understood as an instruction to successively adjoin a category to the root upon insertion of new nodes in the derivation, irrespective whether the category is feature or type incompatible with its sister node. Thus, this final revision enables the Survive principle to also react to type mismatches, essentially subsuming the work of the TDI. Moreover, by stating TSP as in (9), it becomes possible to recognize the non-accidental similarities between syntactic displacement and TDI. On the present conception, these two conditions manifest the syntactic and the semantic side of a single, more fundamental principle, which expels semantically or syntactically incompatible categories from their position.

But the current way of treating type mismatches also differs in an important respect from the conjunction of the standard definition of TDI and the original TSP in (4). This disparity manifests itself, among others, in configurations such as (6a) (repeated below as (10a)), in which a generalized quantifier denoting category α is merged with a type incompatible head β. For such environments, (8) generates a prediction which distinguishes it from the combined effects of the original TSP in (4) and TDI. In contrast to the original TSP in (4) or TDI, (8) leads one to expect that the derivation creates intermediate landing sites for α not only *outside* βP, but also in *βP-adjoined* positions (marked by ➜):

(10) a. Type incompatibility



b.



More specifically, TSP licenses movement through βP in (10b) by having α strand an individual type trace, which may directly combine with the verb denotation. (The entire chain including the quantifier finally becomes interpretable once α attaches to a node of type t in a later movement step, also shown in (10b)). Crucially, such an intermediate position could not have been generated by TDI, because the lexical content of α is not interpretable in the intermediate βP-adjoined position. A generalized quantifier denotation cannot combine with a two-place relation without type adjustment. Actual manifestations of the structure (10b) will be encountered in Section 3 (see (20)[7]).

Two concluding remarks on the present implementation of the TSP are in order here. First, although (9) superficially differs substantially from the original formulation of TSP (1)/(4), it strictly adheres to the spirit of the Survive program, which regulates the distribution of push chains in syntax. Moreover, from a methodological perspective, the generalized version facilitates the detection of a wider range of potential Survive phenomena, rendering it thereby better suitable for the search of criteria that help to discriminate between survive and Attract based models. Second, the current definition of TSP, on which movement always targets the next branching node in the tree, renders storing elements in the WorkBench, and retrieving them again, as in Stroik (2009) unnecessary (see Stroik 2009 for discussion). This significantly simplifies the way in which TSP treats movement.

To summarize, by eliminating stipulations from the original version of the TSP, all movements of feature or type incompatible categories proceed now via local adjunction to the minimally containing node. The final version of TSP in (9), which also subsumes the effects of TDI, accordingly represents a more general, and simpler, manifestation of Survive, which both extends the range of categories affected by TSP and the set of possible landing sites for movement. These two properties will be seen to receive empirical support from the account of scope restrictions to be given in Section 3, where they will prove instrumental for a unified analysis of the phenomena.

2.2 The phonological theory of QR

A second component that is crucial to the analysis to be developed in Section 3 is provided by what has come to be known as the *phonological theory of QR*. On

---

7. So far, I have only been able to find indirect evidence for such intermediate positions. Potential *prima facie* support for the presence of VP-adjoined traces come from licensing of bound variable readings by quantificational objects into low VP-adjuncts, as in (i):

(i) John read every book before reviewing it.

For (i), it might be argued that the variable is licensed by a VP-adjoined e-type trace of the object.

this view, QR does not proceed in a separate, covert level of representation which follows overt syntax, but takes place in the stem of the derivation and interleaves with other, overt movement operations (Bobaljik 1995; Fox & Nissenbaum 1999; Bhatt & Pancheva 2004). The fact that QR does not affect the relative order of constituents is accounted for by the assumption that dislocation by *overt covert movement* (Pesetsky 1998) leads to pronunciation of the tail, instead of the head, of the chain.

The phonological theory of QR offers both empirical as well as conceptional advantages over the standard LF-based model. As for the latter, it is no longer necessary to postulate a separate post-Spellout portion of the derivation. This is of advantage as the existence of such a level has commonly been inferred on the basis of the observation that LF behaves just like overt movement, with the only exception that QR does not have a PF-visible effects. But then, the question arises of why to locate these silent dislocation operations into a separate component in the first place. An integrative model, which employs overt covert movement eliminates this redundancy, rendering the transition from syntax to the semantic component more transparent.

Empirically, the phonological model of QR sheds new light, among others, on the interaction between binding and coreference on the one hand, and dislocation on the other. Fox and Nissenbaum (1999) note, for instance, that overt covert movement offers a ready explanation for the observation that extraposition bleeds disjoint reference effects (Taraldsen 1981):

(11) a. *I showed him$_3$ a book [that Sam$_3$ wanted to read] yesterday
b. I showed him$_3$ a book *t* yesterday [that Sam$_3$ wanted to read]

According to Fox and Nissenbaum, extraposition consists in a two step procedure: overt covert movement of the head of the relative clause *a book*, illustrated by (12a), followed by Late Merge of the relative clause to the head, as detailed in (12b).

(12) I showed him$_3$ a book *t* yesterday [that Sam$_3$ wanted to read]
a. Step 1, overt covert movement of *a book*:
I [$_{VP}$ showed him$_3$ a book] yesterday [~~a book~~]
b. Step 2, Late Merge of relative clause:
I [$_{VP}$ showed him$_3$ a book] yesterday [~~a~~ [~~book~~][that Sam$_3$ wanted to read]]

The combination of overt QR and late merge of the relative results now in a derivation in which the name inside the relative clause has never been inside the c-command domain of the coreferential pronoun. Crucially, this analysis is dependent on the assumption central to the phonological theory of QR that inaudible and overt movement processes behave alike: both apply in the stem of the derivation, and both may serve as attachment sites for late merged relative clauses.[8]

3. Survive and scope

In the present section, it will be demonstrated that given the assumptions specified in Section 2, a theory which includes TSP finds support from two different empirical domains related to quantifier scope interpretation.

To be specific, TSP makes possible a simple analysis of two restrictions on relative quantifier scope in English to be discussed in sections 3.1 and 3.2 below, respectively. Although both of these restrictions have already been given detailed analyses in Attract based frameworks (see Bruening 2001 & Sauerland 2000), these accounts were forced to adopt unnatural assumptions, in particular about the interpretation of subjects. Building on these studies, I propose that certain aspects of the Survive model lead to an improvement over these previous accounts in two regards. More specifically, additional machinery will be seen to be superfluous in the TSP model, where all dislocation processes – irrespective whether they affect subjects or objects – proceed in small incremental steps.

3.1 Scope freezing in double object constructions

The conditions regulating possible quantifier scope permutations in English determine that in a number of contexts, the scope order must not alter surface order of the quantificational terms. Prominently among these is the scope freezing generalization for the double object construction (Barss & Lasnik 1986; Richards 2001a; Bruening 2001).

3.1.1 *Previous accounts*

The scope freezing generalization for double object constructions makes explicit the observation that in the dative-accusative serialization, the two internal arguments are unable to change their relative scope order (ibid.):

(13) a. I gave a child each doll. $\exists \succ \forall / {}^*\forall \succ \exists$ (Bruening 2001, (2a))
b. The judges awarded a (#different) athlete every medal.
$^*\forall \succ \exists$ (ibid., (28c))

In the same context, the direct object may scope over the subject, though (Bruening 2001), as shown by (14). This indicates that the relevant constraint does not limit

8. Additional evidence in favor of a single output model and discussion of the phonological theory of QR can be found in Bhatt & Pancheva (2004); Bobaljik (1995); Fox (2002); Hulsey and Sauerland (2003); Nissenbaum (2000); Pesetsky (1998) and Takahashi (2006).

the scope taking options of direct objects *per se*, but has to be formulated in such a way that it imposes order preservation on the relation between the two internal arguments:

(14) a. A (different) teacher gave me every book. $\forall \succ \exists$ (Bruening 2001, (28))
b. At least two judges awarded me every medal. $\forall \succ$ *at least two*

According to Bruening, scope freezing is a reflex of two properties: First, quantificational DPs need to check a Q(uantificational)-feature on v^0. On this view, QR is not only motivated by TDI, but also feature driven. Second, there is a general syntactic requirement that multiple movements to a single position result in tucking in((Richards 2001a, 2001b) and the formation of crossing paths. The latter condition can be further reduced to general principles of economy (Shortest Move and Shortest Attract). Without going into the details (see Richards 2001b), the essential property of the tucking-in approach consists in an isomorphism from order of movement to asymmetric c-command relations. Whenever two categories α and β are attracted by a single head and α moves prior to β, then α asymmetrically c-commands β in the resulting output representation.

Taken together, these two assumptions derive scope freezing for (13). The indirect object (IO) is base generated in a position above the direct object (DO), and therefore closer to the attracting head v, which bears an attracting Q-feature (see (15a)). IO accordingly undergoes QR first, followed by movement of DO. Both QPs land in an outer specifier of vP, as shown in (15b). Moreover, since the order of movement matches the relative order of the landing sites, the two QPs are raised in a crossing dependency, preserving the base order of the internal arguments (see (15c)):

(15) a. $[_{vP}$ SUB $v_{[Q]}$ $[_{VP}$ IO_2 [DO_3]]]
b. $[_{vP}$ IO_2 $[_{vP}$ DO_3 $[_{vP}$ SUB $[_{VP}$ t_2 [t_3]]]]]
c. Mapping from base order to LF: $2 \succ 3 \Rightarrow 2 \succ 3$

Sauerland (2000) notes that Bruening's analysis is challenged by the existence of examples such as (16), which can, among others, be assigned a reading on which the subject scopally interferes inbetween the indirect and the direct object. The relevant parts of the LF are given in (17a):

(16) Two boys gave every girl a flower $\forall \succ 2 \succ \exists$ Sauerland (2000, (49))

(17) a. $[_{vP}$ IO_2 $[_{vP}$ SUB_1 $[_{vP}$ DO_3 $[_{vP}$ t_1 $[_{VP}$ t_2 [t_3]]]]]
b. Mapping from base order to LF: $1 \succ 2 \succ 3 \Rightarrow 2 \succ 1 \succ 3$

(17a) is problematic for the economy/crossing dependencies approach inasmuch as the LF representation does not preserve the order between IO and SUB (see (17b)). While SUB c-commands the IO in the base, the relations are reversed subsequent to QR.

Sauerland proposes to amend this shortcoming by adopting a definition of closeness on which the reconstructed subject and the indirect object occupy two positions at LF which are equidistant to v^0. Together with the assumption that both IO and SUB need to check a Q-feature on v^0, either IO or SUB can be the first category attracted by v. Since order of movement translates into asymmetric c-command, both scope orders SUB $\succ$ IO and IO $\succ$ SUB can now be generated.

Attract analyses need to espouse two rather unnatural propositions, though. First, subsequent to reconstruction, subjects need to undergo short QR from their vP-internal base position, despite their being type compatible, and therefore directly interpretable, in SpecvP. Otherwise, it would be impossible to generate the LF-representation (17a), in which the subject is sandwiched inbetween the indirect and the direct object (see Sauerland 2000). Second (and partially related), the assumption of a Q-feature in addition to the principles of TDI partially duplicates the trigger for QR. Concretely, there is only a single case – namely subjects – where the effects of the Q-feature are distinguishable from those of TDI. For object quantifiers, which need to leave VP due to type mismatch anyway, this has the consequence that postulating a Q-feature only obscures the motivation behind QR.

Note finally that Q-features are indispensable for the success of Attract based approaches, and can therefore not simply be dispensed with. More specifically, Q-features are needed for two reasons. First, they are implicated in the creation of crossing paths, thereby ensuring order preservation. And second, a Q-feature is indispensable for driving the subject from its reconstructed base into an intermediate position in (17a). (This follows from the independent assumption that no category can check features in its first merge position). As will be specified below, the Survive based model offers an alternative which does not require QR to be feature driven, and is therefore in a position to avoid these complications.

3.1.2 *Survive and scope freezing*

The greatest strength of TSP lies in its ability to derive the observation that subjects can be evaluated in an intermediate position above SpecvP that is high enough to scope over other operators such as (object) quantifiers:[9]

(18) $[_{vP}$ SUB$_1$ $[_{vP}$ QP/negation/... $[_{vP}$ t_1 ....]]]

This property will be seen to be instrumental in accounting for the scope options of double object constructions.

Earlier, it was assumed that QR proceeds in the overt part of the derivation by overt covert movement. For constructions with quantificational expressions in object position, such as (13), repeated from above, this entails that the VP-internal QPs are expelled in overt syntax, moving successive cyclically to the next higher nodes until they reach a type compatible position (i.e., a position which is sister to a propositional node such as vP).

---

9. For additional justification for interpreting subjects in this intermediate site see Bruening (2001, fn. 25); Johnson & Tomioka (1997); Lechner (2007); Sauerland (2000).

(13) I gave a child each doll. $\exists \succ \forall / {*}\forall \succ \exists$ (Bruening 2001, (2a))

The pertinent double object example (13) involves two internal arguments (instead of only a single one) that need to be evacuated from VP by overt covert movement. This raises a pair of partially related questions, which has been extensively discussed in the literature on Attract, but has received less attention in Survive based models.[10] Which category is expelled by TSP first? and Where do the two QPs land? As will be specified below, the Survive model essentially replies in the same way as theories that employ Attract, the only difference being that the economy metric which underlies the answer to the second question is already part of the definition of TSP.

Addressing order of movement first, a theory that lacks attractors evidently cannot resort to calculating distances to an attracting head. I will therefore assume that the derivation privileges recently merged nodes over categories that have already been merged at an earlier point in the derivation. Since in bottom-up derivations, 'recency' correlates with height of attachment in the (stem of) the tree, higher categories move first. Just like in the Attract based model, this forces IO to move prior to the lower DO in structures like (13) (see also 3.2.2 for discussion).

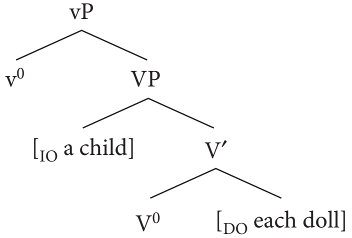
As for the choice of landing sites, the definition of TSP (9), repeated below, already precisely determines the order in which two categories contained inside the same node will land.

(9) *The Survive Principle*
For any nodes α, β and γ, such that γ is the mother of α:
If α is not feature or type compatible with β, then
(i) adjoin α to the result of merging α with β, if β is a head and
(ii) adjoin α to the result of merging γ with β otherwise.

The effects of (9) on the final order of IO and DO are best explicated by going over the actual derivation of (13).

Following Kayne (1984), Beck & Johnson (2006), among many others, the indirect object of double object constructions will be taken to occupy a specifier of the same head that contains the direct object as a complement. Thus, the parse for (13) roughly looks as follows (I use V′ for typographic convenience):

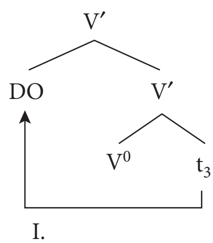
(19)



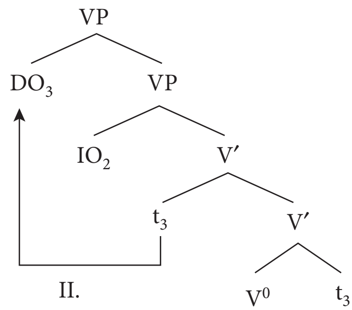
10. See Stroik (2009, chapter 2) for discussion of multiple overt wh-movement.

Assembling the tree for (13) bottom-up, TSP now creates intermediate landing sites for both type incompatible object quantifiers right above every branching node that they are contained in. In (20a), the initial step triggered by (9)(i), DO adjoins to V′, i.e., the minimal node containing V^0 and DO. Next, externally merging IO in SpecVP forces movement of DO across IO, as detailed by (20b) (by (9)(ii)).[11] In (20c), IO then raises across DO to a higher SpecVP in response to the internal merger of DO:

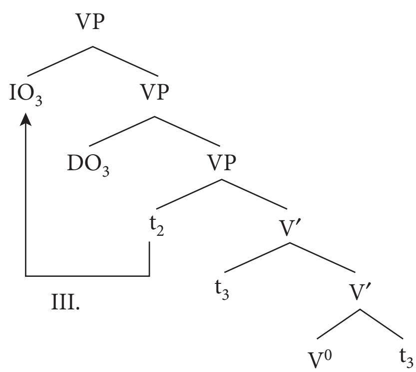
(20) a. Merge V^0, move DO



b. Merge IO, move DO



c. Merge DO, move IO



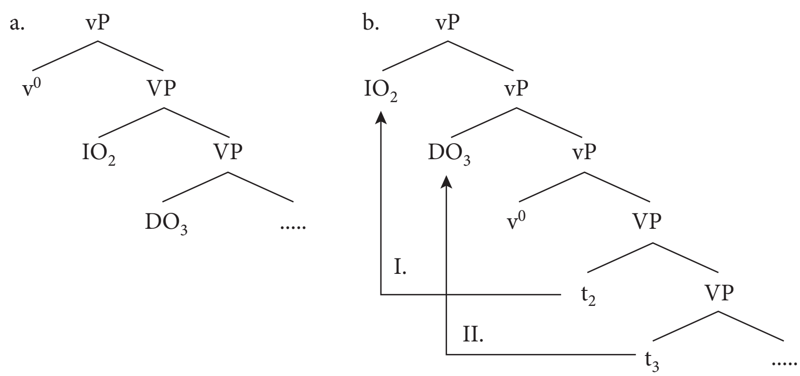
At this point, it becomes apparent that some device is needed in order to terminate TSP-driven movement between IO and DO once the derivation has reached (20c). Otherwise, IO and DO would continue leap-frogging at the VP-level, resulting in endless regress. This is so as in the further course of the derivation, re-merger of DO in a position above IO (as in (20c)) inevitably triggers movement of IO, which in turn pushes DO up one more node derivation, and so on *ad infinitum*.

11. Movement of DO creates a derived λ-abstract. As a result, the V^0-denotation will correctly apply to IO, and not to DO, as its external argument.

A natural strategy for avoiding these types of loops is to take into consideration that multiple locally interleaving steps of the same category as in (20) do not effect a change either in the feature or the type composition of the subtree. Suppose that in a derivation such as (20), DO has crossed over IO and IO over DO (see (20c)), and that neither of these two operations has resulted in improved feature or type compatibility. In this context, moving DO one more time across IO will equally fail to repair feature and type mismatches. Such changes are only expected if a new category is merged inbetween the two object quantifiers. Since the ultimate motivation for TSP resides in the avoidance of feature/type conflicts, and loops of the type discussed above never lead to resolution of these incompatibilities, it is natural to conclude that configurations which create such loops should be generally disregarded by the TSP. Thus, endless regress in (20) is terminated by the independent minimalist core requirement that all movement processes be motivated (*Last Resort*).[12]

Given the proviso above, the derivation of (13) finally arrives at the vP-level, as illustrated by (21a). Addition of v^0 (by clause (ii) of the TSP) creates now intermediate landing sites for IO and DO right above vP. In order to find out which of the two internal arguments moves first, recall that on present assumptions, the order of movement between two nodes is inversely related to the order in which these nodes have been merged. Thus, categories that are merged last will move first. As a consequence, movement of IO across v^0 precedes movement of DO across v^0 in (21b), yielding an order preserving configuration in which IO asymmetrically c-commands DO:

(21)



The computation above demonstrated that whenever two nodes are expelled from a single containing category – in this case the highest segment of vP – the higher

12. This restriction can be built into the definition of movement itself, e.g., by assigning to each category a local memory stack that triggers the instruction to forego movement if two subsequent movement operations are induced by merger of one and the same expression.

one will move first. Then, the lower category moves across the copy of the higher one, followed by a final step, which brings the two nodes again into a local, order preserving relation. Borrowing a term for similar (but not identical) effects from Bobaljik (1995), I will also refer to this interleaving movement dependencies as *leap-frogging*.

Although the present analysis is close in spirit to the traditional Attract account, and empirically replicates the pattern of tucking-in (Richards 2001a, 2001b), it should be noted that there is no reference to 'closeness to an attractor' hidden in any of the definitions or principles underlying the account. Rather, it is the definition of TSP itself, together with the assumption that material that is merged later moves first, which provides the key to the answer for why multiple movements result in crossing, order preserving dependencies. If correct, this conception of the Survive model provides an interesting new perspective on how to derive order preservation effects in syntax.

Finally turning to the derivation of the possible scope orders in the double object construction, recall that while the two internal arguments cannot permute in scope, schematized in (22a), the subject of (16) (repeated from above) may be assigned either intermediate scope inbetween IO and DO (see (22b)) or narrow scope (see (22c)):

(16) Two boys gave every girl a flower $\forall \succ 2 \succ \exists$ Sauerland (2000, (49))

(22) Mapping from base order to scope order:
a. $^*2 \succ 3 \Rightarrow 3 \succ 2$
b. $1 \succ 2 \succ 3 \Rightarrow 2 \succ 1 \succ 3$
c. $1 \succ 2 \succ 3 \Rightarrow 2 \succ 3 \succ 1$

Evidently, the leap-frogging strategy at work in (21), which produces order preserving movement, is the source for the scope freezing effect. However, the representation for (13) given under (21b) is not complete, yet, because (21b) is still missing the subject. This is amended by merging the external argument in (23a):

(23) a. $[_{vP}$ SUB$_1$ $[_{vP}$ IO$_2$ $[_{vP}$ DO$_3$ v^0 ...
b. $[_{vP}$ IO$_3$ $[_{vP}$ SUB$_1$ $[_{vP}$ t$_2$ $[_{vP}$ DO$_3$ v^0 ...
c. $[_{vP}$ IO$_2$ $[_{vP}$ DO$_3$ $[_{vP}$ SUB$_1$ $[_{vP}$ t$_2$ $[_{vP}$ t$_3$ v^0 ...

In (23a), TSP continues to induce displacement of the two internal arguments because the object quantifiers need to combine with type-t expressions, the formation of which in turn requires a subject (trace). Thus, leap-frogging of IO and DO is repeated one more time. First, IO moves over the newly merged subject (23b), followed by movement of DO (23c). (The semantically vacuous movement step of DO across the copy of IO is omitted). In (23c), the two internal arguments have now for the first time reached a position in which they are interpretable as generalized quantifiers, and therefore stop to move.

In the last relevant steps of the derivation, the subject is ejected from its base position and moves to TP. Although the subject matches type-wise with its predicate sister, it is *feature* incompatible with the position it originated in. As a result, it needs to leave SpecvP, and lands, in a first movement step, in a position right inbetween IO and DO:

(23) d. $[_{vP}$ IO_2 $[_{vP}$ SUB_1 $[_{vP}$ DO_3 $[_{vP}$ t_1 v ...

From there, it moves on to SpecTP. Unlike IO and DO, the subject quantifier is interpretable in all intermediate positions, as well as in its base, and is therefore free to undergo optional reconstruction into the slot inbetween IO and DO. If the subject reconstructs into this intermediate position, resulting in an LF-representation akin to (23d), it will be assigned intermediate scope (22a). By contrast, narrow scope of the subject w.r.t. IO and DO (22b) is derived by total reconstruction of SUB into SpecvP.

To recapitulate, The Survive Principle not only accounts for scope rigidity among the internal arguments in the double object construction, but also correctly predicts the more liberal behavior of subjects. The key to the success of the TSP analysis is its ability – in conjunction with the phonological theory of QR – to create interaction between movements driven by type incompatibility and movement driven by feature incompatibility. It is this interaction which sponsors the kind of short subject movement observed in (23d). Moreover, in contrast to the Attract based model, the current analysis derives this result without the need to postulate Q-features, which duplicate the effects of TDI, or stipulating that subject are not interpretable in SpecvP (see Section 3.1.1). If correct, these findings provide support of a Survive based conception of movement, and challenge the traditional Attract model.

Finally, note that the analysis has also consequences for the theory of QR. As discussed in 3.1.1, Fox and Nissenbaum (1999) showed that covert and overt movement both interact with late merge, and should therefore be treated alike. The evidence above supports a second, qualitatively different type of parallelism between covert and overt movement operations, which further vindicates the idea that the two types of processes are not distributed between overt syntax and LF, but take place at one and the same level. More precisely, it was seen that in contexts that involve both overt and covert dislocation, overt movement (of the subject) sometimes seems to follow covert QR (of the objects), instead of preceding it. The order of movement was measured by the possible relative scope orders, which in turn were taken to indicate the availability of intermediate reconstruction sites for the dislocated quantifier phrases. Thus, the findings above not only furnish support for an TSP- based model of dislocation, but also further strengthens the case for a phonological theory of QR.

3.2 Inverse linking

The second freezing phenomenon comes in shape of a scope condition which is operative in context that involve inverse linking, first discussed by Larson (1987) (see also Heim & Kratzer 1998: 234; Sauerland 2000). Restricting the attention to readings in which the embedded QP_3 scopes over its container QP_2, (24) admits a wide and narrow scope reading for the subject, but lacks the intermediate construal (24c):[13]

(24) $[_{QP1}$ Two policemen spy on $[_{QP2}$ someone from $[_{QP3}$ every city]]]
  a. $2 \succ \forall \succ \exists$
  b. $\forall \succ \exists \succ 2$
  c. *$\forall \succ 2 \succ \exists$

For some reason, the subject quantifier may not intervene between the inversely linked quantifier (QP_3) and its container (QP_2). Thus, only two of the three possible mappings from surface to scope order that keep constant the order $QP_3 \succ QP_2$ are actually attested:

(25) Mapping from base order to scope order:
  a. $1 \succ 2 \succ 3 \Rightarrow 1 \succ 3 \succ 2$
  b. $1 \succ 2 \succ 3 \Rightarrow 3 \succ 2 \succ 1$
  c. *$1 \succ 2 \succ 3 \Rightarrow 3 \succ 1 \succ 2$

Sauerland (2000) provides a uniform analysis of this restriction and scope freezing in double object constructions. Although descriptively adequate, the account needs to adopt ancillary assumptions that reduce its explanatory force (see previous section for discussion). Given these inherent shortcomings, I will not further expand on Attract based explanations here, but will directly proceed to the TSP analysis (see Sauerland 2000 for details).

3.2.1 *Survive and inverse linking*

On the TSP account, essential parts of which are graphically represented by (26), insertion of the subject quantifier QP_1 (*two policemen* in (24)) induces movement of the direct object QP_2 (*someone from every city*) to an outer vP-adjoined position. Simultaneously, the embedded category QP3 *every city* is pushed out of its base inside the containing QP_2, landing as a DP-adjunct. Both movements are triggered by type mismatch. At this point of the derivation, there are two categories that could potentially undergo TSP-driven movement. The subject quantifier QP_1, which needs

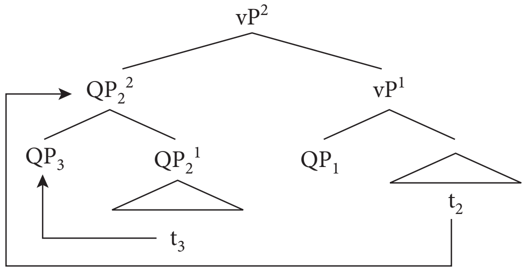
13. Strings of similar structure such as (i) also can be assigned the surface scope order.

(i) Every policemen_1 reviewed exactly two reports_2 about every candidate_3 $(1 \succ 2 \succ 3)$

I assume that this interpretation is derived by interpreting QP_3 *in-situ* (see Heim & Kratzer 1998).

to escape SpecvP to avoid feature clash with its head v°, and QP_3 from its intermediate QP_2-adjoined position, which has to move in order to resolve a type mismatch:

(26)

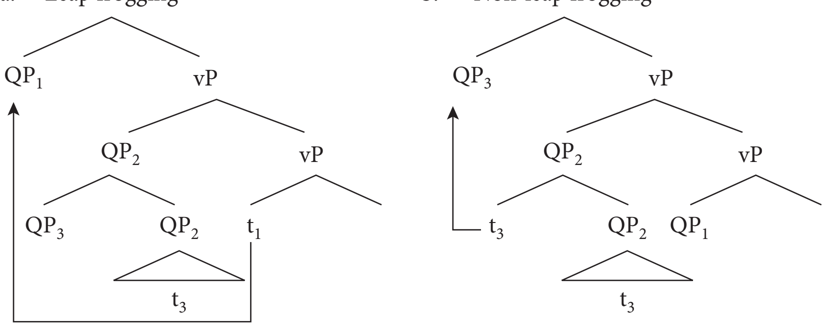


The decision which of these two nodes moves first proves interesting, as it supplies a diagnostic for choosing between the two competing strategies that Attract models and the current inception of TSP employ for ordering multiple dislocations, respectively: closeness vs. leap-frogging. Assume to begin with that precedence of operations were determined on the basis of 'closeness to the target', as is standard practice in Attract based models. Applying this metric to (26), one arrives at an impasse, because QP_3 and QP_1 (the subject) are equidistant to vP^2. Both nodes are separated by exactly one intervening node from the target (QP_3 by a segment of QP_2, labeled $QP_2{}^2$; and QP_1 by the vP-segment vP^1), and in both cases, the intervening node is a segment of a multiple-segment category. Movement of QP_3 and QP_1 should therefore be able to proceed in either order. However, as will be shown below (see discussion of (29)), a system that admits free movement cannot derive scope freezing by any standardly sanctioned conditions.

If on the other side the current conception is correct, on which the order of movement operations is taken to be regulated by leap-frogging, movement of QP_2 above the subject forces the subject to cross over QP_2, as shown by (27a). Moreover, leap-frogging also excludes the alternative derivation (27b) in which movement of QP_2 is immediately followed by subextraction of QP_3, instead of movement by QP_1:

(27)

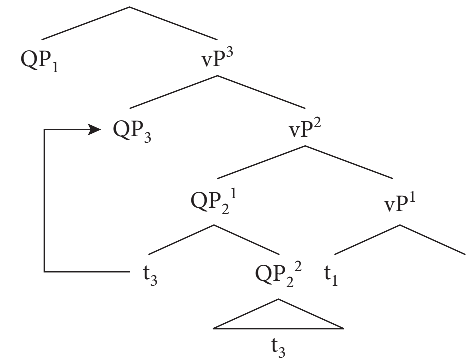
a. Leap frogging b. Non-leap frogging



Adhering to the idea that multiple dislocations are always resolved by leap-frogging, I will assume that (27a) is the correct intermediate representation created by the TSP model.

In the final movement step leading to an interpretable output, the inversely linked QP_3 is expelled from its intermediate position. QP_3 targets the next higher position in the tree, adjoining to vP^2, as detailed by (28). Note on the side that for QP_3, the node that its mother has combined with is vP^1. Hence, QP_3 adjoins to the result of merging QP_2 with vP^1 – i.e., vP^2 – and not to some position above QP_1.

(28)



In (28), all quantifiers have now reached a position in which they are interpretable (the subject will move on to SpecTP, but this is irrelevant for present concerns). What is of particular significance is the fact that the subject has not had the option to strand an intermediate trace inbetween QP_3 and QP_2. As a result, the subject can be read with widest possible scope, yielding reading (25a), or narrow scope, as in (25b), but cannot be construed with intermediate scope, as in (25c).

Thus, TSP provides a simple explanation for scope freezing in inversely linked contexts. The key ingredient of the analysis is again order of movement. Although feature driven subject movement and QR both proceed at the same level, that is in overt syntax, subject displacement precedes inverse linking on the assumption that derivations follow leap-frogging.

3.2.2 *Order of movement*

One open question remains to be addressed. What is the defining property of leap-frogging? Can it be derived from another axiom of the system? One venue to conceptualize the underlying principle governing these effects, which made reference to order of merge, was already explored in the discussion of double object constructions. There, it was concluded that categories which are merged last move first. For instance, in the double object case (20), the higher IO, which was added to the derivation later than the lower DO, underwent movement first. And in (27), the subject is merged at a later point in the derivation than the inversely linked object QP_3 and therefore raises before QP_3. On this view, the metric determining movement orders recycles an independent property of movement, i.e., the order of external merge.

I will end this section with a brief discussion of a related issue that arises in the analysis of inverse linking. As mentioned above, the order of merge informs order of movement. Interestingly, this metric only uses information about external merge operations in (27), while being ignorant about applications of internal merge or remerge. If the order in which categories are internally merged would also count, then derivation (27b) could no longer be excluded. More specifically, suppose that adjunction of the inversely linked QP_3 to its container QP_2 is delayed to a point *after* QP_2 has moved to vP, as detailed by the transition from (29a) to (29b). Then, the node that has been merged last would be QP_3, and QP_3 should therefore move prior to the subject QP_1, resulting in the intermediate representation (29c):

(29) a. $[_{vP}$ $[_{QP2}$ ....QP_3 .....] $[_{vP}$ QP_1 (result of movement of QP_2)

b. $[_{vP}$ $[_{QP2}$ QP_3 $[_{QP2}$ ....t_3 .....]] $[_{vP}$ QP_1 (adjoin QP_3 to QP_2)

c. $[_{vP}$ QP_3 $[_{vP}$ $[_{QP2}$ t_3 $[_{QP2}$ ....t_3 .....]] $[_{vP}$ QP_1 (move QP_3 to vP)

d. $[_{vP}$ QP_3 $[_{vP}$ QP_1 $[_{vP}$ $[_{QP2}$ t_3 $[_{QP2}$ ....t_3 .....]] $[_{vP}$ t_1 (move QP_1 to vP, resulting in $QP_3 \succ QP_1 \succ QP_2$)

But in such a scenario, the TSP forces the subject to stop inbetween QP_2 and QP_3, as in (29d), feeding the unattested scope order $QP_3 \succ QP_1 \succ QP_2$. Thus, internal merge operations such as movement of QP_3 in (29b) must be somehow rendered invisible for the calculation of movement orders in (27b).

As far as I can see, there are two ways to exclude (27b) and similar derivation. On the one hand, it is possible to explicitly exclude prior movement operations from determining the order of future derivational steps. Although not implausible, this approach only yields suboptimal results. First, it fails for the double object cases, where movement of DO across IO forces the latter to raise. Second, it introduces an imbalance between internal and external merge, in that movement orders would only be dependent on properties of the latter. Finally, this view presupposes the availability of non-local information flow, because the nodes have to 'remember' throughout the derivation the order in which they originated.

Alternatively, one might invoke an independent property for blocking the derivation (27) above, while at the same time keeping the assumption that the order of movement is a function of the history of external as well as internal merge operations. A likely candidate for such a restriction is provided by an earliness requirement of the sort employed by Pesetsky (1989) and Stroik (2009). If operations need to be carried out at the earliest possible point in the derivation, QP_3

has to adjoin to its container QP_2 even before the container has reached the vP-adjoined position in (27). It follows that the category which is merged last is not QP_2, but the subject, and the derivation therefore proceeds, as desired, along the lines of (27a). In contrast to the first option, it is not necessary to have the relation only being defined for instances of external merge. Also, this conception, which will be adopted here, has the added advantage that introducing non-local dependencies can be avoided.

In sum, linking the derivational history of categories to the timing of future derivational steps represents a promising method for ordering movements. Taken together, the system employs two distinct timing metrics. On the one hand, if two (or more) nodes are to undergo TSP induced movement at a specific step in the derivation, then the category that has been merged last will be expelled first. On the other hand, competition between movement processes that cannot be ordered by this metric, e.g., because these processes are part of different subderivations (see inverse linking), is indirectly resolved by the requirement that operations apply as early as possible. At this point, it remains to be seen whether either one of these conditions can be reduced to the other, or whether they can be derived from a common underlying property. In addition, some obvious questions need to be answered. Most importantly, one would like to find a plausible reason why order of merge and order of movement should be linked in such an intimate way, and how this relation can be formally implemented. I will have to delegate these issues to future investigations.

4. Conclusion

In this paper, I tried to identify ways to discriminate between the predictions of an Attract based and a survive based model of non-visible movement operations (more concretely, QR). The discussion lead to four broader conclusions.

To begin with, rephrasing the TSP in such a way that it generates intermediate position on top of every node a moving category has to cross leads to a definition that is less stipulative, and at the same time naturally subsumes the (methodologically and substantially) closely related principle of Type Driven Interpretation.

Second, the 'push-chain'-TSP based perspective on movement receives support from the fact that two conditions limiting scope taking directly follow from the restrictions that TSP imposes on the way in which feature driven subject raising interacts with QR.

Third, the very existence of such interactions between QR and subject movement can be taken as an indication that both dislocation processes apply at the same level of representation, so as to permit the derivational interactions discussed in Section 3 above. This finding supplies further evidence vindicating the phonological theory of QR.

Finally, only a push chain model, in which moving categories are expelled instead of attracted, leads one to expect that movement dependencies which involve such fundamentally different elements as subjects (bearing feature-incompatible subject features) and quantifiers (which are type-incompatible) systematically interact with each other. The standard Attract theory has to express this conspiracy indirectly, by positing features on quantifiers and a special proviso for the interpretation of subjects. On the Survive model, the interdependencies between QR and subject movement fall out from a basic property of the theory, i.e., the assumption that all feature or type incompatible categories are removed from their local environment, irrespective of the nature of the incompatibility.

If the above conclusions are essentially accurate, contexts involving different types of movements elicit important empirical diagnostics for evaluating the respective strengths of the two competing models, rendering such phenomena interesting candidates for future explorations into the Survive program.

References

Barss, A. & Lasnik, H. 1986. A note on anaphora and double objects. *Linguistic Inquiry* 17: 347–354.

Beck, S. & Johnson, K. 2004. Double objects again. *Linguistic Inquiry* 35(1): 97–124.

Bhatt, R. & Pancheva, R. 2004. Late merger of degree clauses. *Linguistic Inquiry* 35(1): 1–45.

Bobaljik, J. 1995. Morphosyntax: The syntax of verbal inflection. Ph.D. dissertation, MIT.

Bruening, B. 2001. QR obeys superiority: Frozen scope and ACD. *Linguistic Inquiry* 32(2): 233–273.

Chomsky, N. 1995. *The minimalist program* [Current Studies in Linguistics]. Cambridge MA: The MIT Press.

Chomsky, N. 2005. On phases. Ms, MIT.

Fox, D. 2002. Antecedent contained deletion and the copy theory of movement. *Linguistic Inquiry* 33(1): 63–96.

Fox, D. & Nissenbaum, J. 1999. Extraposition and scope: A case for overt QR. In *WCCFL 18*.

Heim, I. & Kratzer, A. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Hulsey, S. & Sauerland, U. 2003. Sorting out relative clauses: A reply to Bhatt. Ms, University of Massachusetts at Amherst.

Johnson, K. & Tomioka, S. 1997. Lowering and mid-size clauses. In *Proceedings of the Tübingen Workshop on Reconstruction*, G. Katz, Shin-sook Kim & Winhart, H (Eds), 185–206. Tübingen.

Kayne, R. 1984. Unambiguous paths. In *Connectedness and binary branching*, 129–164. Dordrecht: Foris.

Larson, R. 1987. Quantifying into NP. Ms, MIT.

Lechner, W. 2005/07. Interpretive effects of head movement. Available for download at http://ling.auf.net/lingbuzz.

Nissenbaum, J. 2000. Investigations of covert phrase movement. Ph.D. dissertation, MIT.

Pesetsky, D. 1989. Language-particular processes and the earliness principle. Ms, MIT.

Pesetsky, D. 1998. *Phrasal movement and its kin*. Cambridge MA: The MIT Press.

Preminger, O. 2008. (Im)perfect domains: Yet another theory of syntactic movement. In *Proceedings of 26th West Coast Conference on Formal Linguistics,* C.B. Chang & H.J. Haynie (Eds), 402–410. Somerville MA: Cascadilla Press.

Putnam, M.T. 2007. *Scrambling and the Survive principle.* Amsterdam: John Benjamins.

Richards, Norvin W. III. 2001a. *Movement in language.* Oxford: OUP.

Richards, Norvin. 2001b. Featural cyclicity and the ordering of multiple specifiers. In *Working minimalism*, S. Epstein & N. Hornstein (Eds), 127–159. Cambridge MA: The MIT Press.

van Riemsdijk, H. 1997. Push chains and drag chains: Complex predicate split in Dutch. In *Scrambling*, S. Tonoike (Ed.), 7–33. Tokyo: Kurosio Publishers.

Sauerland, U. 2000. Syntactic economy and quantifier raising. Ms, University of Tübingen.

Stroik, T. 1999. The Survive principle. *Linguistic Analysis* 282–309.

Stroik, T. 2009. *Locality in minimalist syntax*. Cambridge MA: The MIT Press.

Takahashi, S. 2006. Decomposition and identity. Ph.D. dissertation, MIT.

Taraldsen, K.T. 1981. The theoretical interpretation of a class of marked extractions. In *Theory of markedness in generative grammar: Proceedings of the 1979 GLOW conference*, A. Belletti, L. Brandi & L. Rizzi (Eds), 475–516. Pisa: Scuola Normale Superiore.