A Formal Foundation for A and A-bar Movement

Gregory M. Kobele

University of Chicago
kobele@uchicago.edu

Abstract. It seems a fact that movement dependencies come in two flavours: "A" and "A-bar". Over the years, a number of apparently independent properties have been shown to cluster together around this distinction. However, the basic structural property relating these two kinds of movement, the ban on improper movement ('once you go bar, you never go back'), has never been given a satisfactory explanation. Here, I propose a timing-based account of the A/A-bar distinction, which derives the ban on improper movement, and allows for a simple and elegant account of some of their differences. In this account, "A" dependencies are those which are entered into before an expression is first merged into a structure, and "A-bar" dependencies are those an expression enters into after having been merged. The resulting system is mildly context-sensitive, providing therefore a restrictive account of possible human grammars, while remaining expressive enough to be able to describe the kinds of dependencies which are thought to be manifest.

It is common to describe the syntax of natural language in terms of expressions being related to multiple others, or moved from one position to another. Since [25], much effort has been put into determining the limitations on possible movements. A descriptively important step was taken by classifying movement dependencies into two basic kinds: those formed by the rule `move NP`, and those formed by `move wh-phrase` [7]. This bipartition of movement dependencies is a formal rendering of the observation that wh-movement, topicalization, and comparative constructions seem to have something in common, that they do not share with passive and raising constructions, which in turn have their own particular similarities. Whereas syntactic theories such as Head-Driven Phrase Structure Grammar [22] and Lexical-Functional Grammar [5] have continued to cash out this intuitive distinction between dependency types formally, in terms of a distinction between lexical operations and properly syntactic operations, this distinction has no formal counterpart in theories under the minimalist rubric [9]. This theoretical lacuna has led some [27] to explore the hypothesis that this perceived distinction between movement types is not an actual one; i.e. that the differences between Wh-construction types on the one hand and passive construction types on the other are not due to differences in the kinds of dependencies involved. A problem besetting those minimalists eager to maintain the traditional perspective on the difference between wh- and NP-movement

C. Ebert, G. Jäger, and J. Michaelis (Eds.): MOL 10/11, LNCS 6149, pp. 145–159, 2010.

dependencies, is that there is no principled distinction between long-distance dependency types available in the theory; a theory with one kind of long-distance dependency does not lie well on the procrustean bed of one with two. The contribution of this paper is to provide a non-*ad hoc* minimalist theory with two kinds of movement dependencies, which have just the kind of properties which have become standardly associated with `move NP`- and `move wh-phrase`-related phenomenon, respectively. It is important to note that it is not a particular analysis of a particular language which I will claim has these properties, but rather the *theoretical framework* itself. Once we are in posession of a theoretical framework in which we have two movement dependency forming operations that interact in the appropriate way, we are in a position to determine whether the old intuitions about movement dependencies coming in two types were right; we can compare the relative elegance of *analyses* written in one framework to those written in the other.

In §1 I describe the kinds of properties which are constitutive of the empirical basis for the bifurcation of movement into two types. Recent minimalist accounts of some of these properties [16,4] will form the conceptual background of my own proposal, developed in §2. The formal architecture of minimalist grammars [28] is presented in §2.1, and it is extended in §2.2 in accord with my proposal. In §2.3, I present an analysis of passivization in English (drawing on the smuggling account proposed in [10]) written within the framework of §2.2.

1 On A and A-bar Movements

Many differences between NP and wh-phrase movement have been suggested in the literature, such as whether they license parasitic gaps, whether they can move out of tensed clauses, whether they bar the application of certain morpho-phonological processes, and whether they incur crossover violations (see e.g. [18]). (These questions uniformly receive a negative answer with respect to NP movements, and a positive one with respect to wh-phrase movements.) A perusal of these properties makes clear that they are highly construction- and analysis-specific. In other words, a theoretical framework cannot derive these differences between NP and wh-phrase movement *simpliciter*, but may at most derive them relative to particular analyses of these constructions. The only *analysis independent* property of NP and wh-phrase movement types is the so-called 'ban on improper movement', which states that NP movement of an expression may not follow its movement as a wh-phrase. This relational property of NP (henceforth: 'A') and wh-phrase (henceforth: 'A-bar') movements is widely accepted, and was motivated by the desire to rule out sentences such as (1) below.

(1). *[$_S$ John seems [$_{\overline{S}}$ t [$_S$ t wanted to sleep]]]

In (1), the first movement (to SPEC-$\overline{S}$) is an A-bar movement, and the second (to the matrix clause subject position) an A movement. The unacceptability of (1) contrasts with the well-formed (2) below, which one can interpret as suggesting that it is the second movement in (1), from SPEC-$\overline{S}$ to the subject position in the matrix clause, which leads to the deviance of (1).

(2). $[_{\overline{S}}$ Who does $[_S$ Mary believe $[_{\overline{S}}$ $\mathtt{t}$ $[_S$ $\mathtt{t}$ wanted to sleep]]]]

In the government and binding (GB) framework (as described in [8]) and the minimalist program (MP) (as in [9]), the ban on improper movement must simply be stated as such; movement from an A position may target an A-bar position, but movement from an A-bar position may only target other A-bar postions (see [21] for a particularly articulated view). In LFG and HPSG, where A movements are taken to be resolved lexically and A-bar movements resolved grammatically, the ban on improper movement follows from the architecture (grammatically complex expressions are simply not the kinds of things that lexical processes apply to). In the grammatical architecture I will develop in §2, A movements are those which occur before, and A-bar movements those which occur after, an expression has been first merged into a structure. The ban on improper movement is then just a simple consequence of the structure of derivations.

Strictly speaking, the ban on improper movement is the only property of movement types which a grammatical framework can be said to derive. However, the following more analysis-specific property of movement types listed above will be shown to follow naturally from the architecture of the system in §2.

A and A-bar movements differ systematically as to whether they create new binding possibilities.[1] Consider sentences (3) and (4) below. In (3), the reflexive pronoun *himself* cannot be bound by the quantified noun phrase *every boy*, whereas in (4), after movement, it can.

(3). *It seems to himself that every boy is wonderful.
(4). Every boy seems to himself to be wonderful.

This situation contrasts with the one laid out in (5) below, where we see that a wh-moved expression is not able to bind the reflexive pronoun. Sentence (6) shows that it is indeed the failed attempt at binding that results in the ungrammaticality of (5), as this movement is otherwise fine.

(5). *Which boy does it seem to himself that Mary loves?
(6). Which boy does it seem that Mary loves?

The difference between these movement types can be summed up in the following diagram, with *A* movement of *XP* being able to, while *A-bar* movement of XP being unable to, bind the pronoun *pro*:

$$XP \ldots [\ldots \mathit{pro} \ldots \mathtt{t} \ldots]$$

Attempts to account for these phenomena have been numerous in the GB framework, and have continued into the MP (see [26] for an accessible typology). One option is to rule out rebinding by A-bar movements by denying the

[1] This is called 'crossover' in the literature [23]. Strong crossover is when the bound expression c-commands the source position of the movement, and weak crossover is when the bound expression is properly contained in such a c-commanding phrase. Weak crossover violations have been argued to be ameliorable under certain conditions [17].

ability to bind pronouns from A-bar positions, and another is to require that no closer potential binders may intervene between an A-bar trace and its antecedent. Given the framework developed below, we are in a position to stipulate that an expression may bind only those expressions that it c-commands when first merged (in other words, that binding is determined by c-command in the derivation tree).

2 Trace Deletion, Derivationally

Without a formal apparatus to hang an account of differences between movement on, researchers in the GB tradition have attempted to capture the difference between A and A-bar movements in terms of properties of source positions: traces. It was discovered that under a certain network of assumptions, A-bar traces behaved as R-expressions, and A traces as anaphors. In the MP, it has been suggested for diverse reasons that A-bar traces should be treated formally as copies of the moved expression, while A traces should be treated formally as unstructured objects [11,16]. This is the idea upon which this paper builds. But currently there is nothing more than arbitrary stipulation (why are some traces copies, and others not? why does the ban on improper movement hold?). To excavate the idea, we should get clear on what, exactly, traces are (for).

In mainstream minimalism, movement chains are licensed derivationally: only well-formed chains are built in the first place. Therefore, traces are not needed for evaluating well-formedness of a syntactic representation (their role in government and binding theory). Instead, traces (qua copies) play a role primarily at the interfaces, in particular the syntax-semantics interface, where they determine the positions in which an expression may take scope, as per [9]. The distinction between structured and unstructured traces (i.e. copies versus 'traditional' traces) is intended to indicate the possibility or not of reconstruction (with expressions being reconstructible into structured trace positions, but not into unstructured trace positions).

A 'copy' indicates that an expression is present at a particular location in the structure for the purposes of reconstruction, while (unstructured) traces indicate that it is not. The intuition is simply that an expression may be interpreted in any position in which it is present; it is present in its A-bar positions, but not (necessarily) in its A positions. This is easier to understand if we think not about derived structures, but about the derivation itself: talk of 'copies' versus 'traces' is recast in terms of whether (copies) or not (traces) the object which is entering into these various dependencies is already present in the derivation at the time the dependency in question is entered into.

The basic formal idea behind this intuition is to incorporate both transformations, as well as 'slash-feature percolation' [13] into a single formalism. Then we may have derivations involving slash-features, in which the object entering into the dependency in question is not present at the time the dependency is established:

1. $[_V$ write $]$
2. $[_{VP/DP}$ was written $]$

In addition, we may have derivations using transformations, in which the object entering into the dependency in question *is* present at the time the dependency is established:

1. $[_{S'}$ that $[_S$ book was written $]$ $]$
2. $[_N$ book $[_{S'}$ that $[_S$ t was written $]$ $]$ $]$

The present derivational reconstruction of the representational traces versus copies account of the A/A-bar distinction has the distinct advantage of giving a unified and intuitive account of various properties of A and A-bar movement. In particular, the ban on improper movement is forced upon us in this timing-based perspective on long-distance dependency satisfaction.

In the next section I show how to incarnate this derivational perspective on A and A-bar movement in a formal system. In so doing we gain a better understanding not only of the mechanisms involved, but also of the various analytical options which the mechanisms put at our disposal.

2.1 Minimalist Grammars

Minimalist grammars [28] provide a formal framework within which the ideas of researchers working within the minimalist program can be rigorously explored. A minimalist grammar is given by a four-tuple $\langle V, Cat, Lex, \mathcal{F}\rangle$, where

- V, the alphabet, is a finite set
- Cat, the set of features, is the union of the following pair of disjoint sets:
  - $\mathbf{sel} \times \mathbf{Bool}$, where for
    - $\langle x, 0\rangle \in \mathbf{sel} \times \mathbf{Bool}$, we write `=x`, and call it a *selector* feature
    - $\langle x, 1\rangle \in \mathbf{sel} \times \mathbf{Bool}$, we write `x`, and call it a *selectee* feature
  - $\mathbf{lic} \times \mathbf{Bool}$, where for
    - $\langle y, 0\rangle \in \mathbf{lic} \times \mathbf{Bool}$, we write `+y`, and call it a *licensor* feature
    - $\langle y, 1\rangle \in \mathbf{lic} \times \mathbf{Bool}$, we write `-y`, and call it a *licensee* feature
- Lex, the lexicon, is a finite set of pairs $\langle v, \delta\rangle$, for $v \in V \cup \{\epsilon\}$, and $\delta \in Cat^*$
- $\mathcal{F} = \{\mathbf{merge}, \mathbf{move}\}$ is the set of structure building operations

Minimalist expressions are traditionally given in terms of leaf-labelled, doubly ordered (projection and precedence) binary trees. The leaves are labelled with pairs of alphabet symbols ($V \cup \{\epsilon\}$) and feature sequences (Cat^*). A typical expression is given in figure 1, where the precedence relation is indicated with the left-right order, and the projection relation is indicated with less-than ($<$) and greater-than ($>$) signs.

The projection relation allows for the definition of the important concepts 'head-of' and 'maximal projection'. Intuitively, one arrives at the leaf which is the head of a complex expression by always descending into the daughter which is least according to the projection relation. In the tree in figure 1, its head is

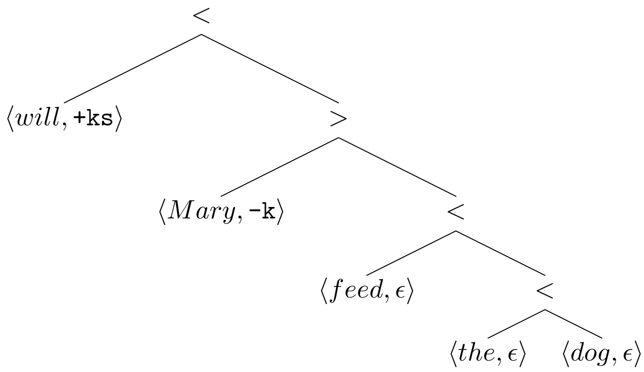

Fig. 1. A minimalist expression

$\langle will, \texttt{+ks} \rangle$, which is also (trivially) the head of its root's left daughter. The head of the root's right daughter is $\langle feed, \epsilon \rangle$. Given a tree t with head $\langle v, \delta \rangle$, we write $t[\delta]$ to indicate that the head of t has features δ. A proper subtree t' of tree t is a maximal projection just in case the sister t_s of t' is such that $t_s < t'$ in t. If t' is a subtree of a tree t, we may write t as $C\langle t' \rangle$. $C\langle t'' \rangle$ then refers to the tree like t but with the subtree t' replaced by the subtree t''.

Work by [19] has shown that the operations of **merge** and **move** can be completely supported by data structures far less structured than doubly ordered leaf labelled binary trees.[2] Accordingly, [29] provide a simplified expression type for minimalist grammars; an expression is a sequence $\phi_0, \phi_1, \ldots, \phi_n$, where each ϕ_i is a pair $\langle \nu, \delta \rangle$, for $\nu \in V^*$, and $\delta \in Cat^+$. The intuition is that each ϕ_i, $1 \leq i \leq n$, represents the phonetic yield of a moving subtree, and that ϕ_0 represents the phonetic yield of the rest of the tree.

Let $t_1[\texttt{=x}\delta_1]$ and $t_2[\texttt{x}\delta_2]$ be two minimalist trees with head-features beginning with =x and x respectively. Then the result of merging together $t_1[\texttt{=x}\delta_1]$ and $t_2[\texttt{x}\delta_2]$ is shown in figure 2.[3]

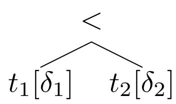

Fig. 2. **merge**$(t_1[\texttt{=x}\delta_1], t_2[\texttt{x}\delta_2])$

From the perspective of the more concise chain-based representation, **merge** is broken up into two subcases, depending on whether or not the second argument

[2] [15] has shown that these trees are also unnecessary for semantic interpretation.

[3] The **merge** operation presented here is non-standard in that it only allows for merger into a complement position (i.e. the merged expression follows the expression to which it is merged). I adopt this simplification only for expository purposes; nothing important hinges on this.

will move (i.e. whether $\delta_2 = \epsilon$ or not).

$$\mathbf{merge1}(\langle \nu_1, \texttt{=x}\delta_1 \rangle, \phi_1, \ldots, \phi_m;\ \langle \nu_2, \texttt{x} \rangle, \psi_1, \ldots, \psi_n) = \\ \langle \nu_1 \nu_2, \delta_1 \rangle, \phi_1, \ldots, \phi_m, \psi_1, \ldots, \psi_n$$

$$\mathbf{merge2}(\langle \nu_1, \texttt{=x}\delta_1 \rangle, \phi_1, \ldots, \phi_m;\ \langle \nu_2, \texttt{x}\delta_2 \rangle, \psi_1, \ldots, \psi_n) = \\ \langle \nu_1, \delta_1 \rangle, \phi_1, \ldots, \phi_m, \langle \nu_2, \delta_2 \rangle, \psi_1, \ldots, \psi_n$$

Let $C\langle t[\texttt{-y}\delta_2]\rangle[\texttt{+y}\delta_1]$ be a minimalist tree with head features beginning with `+y`, which contains a maximal (wrt projection) subtree $t[\texttt{-y}\delta_2]$ with head features beginning with `-y`. Then the result of applying the **move** operation to $C\langle t[\texttt{-y}\delta_2]\rangle[\texttt{+y}\delta_1]$ is shown in figure 3 (where $\lambda = \langle \epsilon, \epsilon \rangle$).

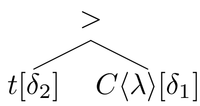

Fig. 3. $\mathbf{move}(C\langle t[\texttt{-y}\delta_2]\rangle[\texttt{+y}\delta_1])$

Turning once more to the more concise chain-based representation, **move** is broken up into two subcases, depending on whether or not the moving subtree will move again (i.e. whether $\delta_2 = \epsilon$ or not).

$$\mathbf{move1}(\langle \nu_1, \texttt{+y}\delta_1 \rangle, \phi_1, \ldots, \langle \nu_2, \texttt{-y} \rangle, \ldots, \phi_m) = \\ \langle \nu_2 \nu_1, \delta_1 \rangle, \phi_1, \ldots, \phi_m$$

$$\mathbf{move2}(\langle \nu_1, \texttt{+y}\delta_1 \rangle, \phi_1, \ldots, \langle \nu_2, \texttt{-y}\delta_2 \rangle, \ldots, \phi_m) = \\ \langle \nu_1, \delta_1 \rangle, \phi_1, \ldots, \langle \nu_2, \delta_2 \rangle, \ldots, \phi_m$$

Since at least [25] it has been observed that movement cannot relate arbitrary tree positions, but rather that there are constraints on which positions a moved item can be construed as originating from. The canonical constraint on movement in minimalist grammars is the SMC [28], intended to be reminiscient of the *shortest move constraint* of [9].[4] Intuitively, the SMC demands that if an expression *can* move, it *must* move. This disallows cases in which two or more moving subexpressions 'compete' for the same `+y` feature. The SMC is implemented as a restriction on the domain of **move**:

> $\mathbf{move}(\langle \nu, \texttt{+y}\delta \rangle, \phi_1, \ldots, \phi_m)$ is defined iff exactly one $\phi_i = \langle \nu_i, \delta_i \rangle$ is such that δ_i begins with `-y`

[4] [12] investigate other constraints on movement in minimalist grammars.

2.2 Trace Deletion in Minimalist Grammars

Minimalist grammars as presented above allow only for 'derivational copies'; an expression is present in the derivation at every point in which it enters into a syntactic dependency. This is because we first merge an expression into the derivation, and then satisfy further dependencies by moving it around. In order to allow for 'derivational traces', we need an expression to start satisfying dependencies before it is part of the derivation. The mechanism adopted to allow this somewhat paradoxical sounding state of affairs bears strong similarities to 'slash feature percolation' in GPSG, as well as to hypothetical reasoning in logic. The intuition is that we will allow ourselves, upon encountering an expression $t[\langle z, 0\rangle\delta_1]$, to *assume* the existence of an expression with a matching feature $\langle z, 1\rangle$. This allows us to continue the derivation *as if* we had successfully checked the first feature of t. However, assumptions, like other forms of credit, must eventually be paid back. This takes here the form of inserting an expression which actually has the features we had theretofore assumed we had, *discharging* the assumptions.

To implement hypothetical reasoning, we introduce another pair of operations, **assume** and **discharge**. Informally, **assume** eliminates features of an expression, and keeps a record of the features so eliminated. An example is given in figure 4, where $\mathtt{t_d}$ represents the information that a $\mathtt{d}$ feature was hypothesized.

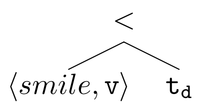

Fig. 4. assume($\langle smile, \mathtt{=d\ v}\rangle$)

To eliminate assumptions, we introduce the **discharge** operation, which 'merges' two expressions together, using the second to satisfy *en masse* some of the features previously eliminated via **assume** in the first. An example is shown in figure 5, where the dotted lines indicate the checking relationships between the connected features.

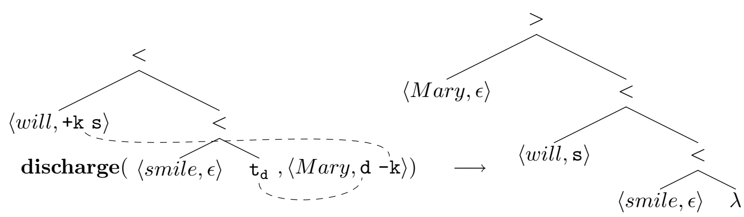

Fig. 5. discharge

While I have described the **assume** operation in terms of hypothesizing one feature away at a time, it is simpler to have the **assume** operation hypothesize an entire feature sequence. We therefore extend the definition of expressions: an expression is a sequence over $(V^* \times Cat^+) \cup (Cat^+ \times Cat^+)$. A subexpression of the form $\langle \delta, \delta' \rangle$, where both $\delta, \delta' \in Cat^+$ indicates a (partially discharged) hypothesis of an expression with feature sequence beginning with $\delta\delta'$. The first component of such a subexpression records which of the hypothesized features have been checked, and the second component which of the hypothesized features remain unchecked (in this sense, a hypothetical subexpression resembles a dotted item in an Earley parser). Accordingly, we define **assume** as per the following:[5] for $\delta_2 \in Cat^+$

$$\textbf{assume}(\langle \nu_1, \texttt{=x}\delta_1 \rangle, \phi_1, \ldots, \phi_m) \rightarrow \langle \nu_1, \delta_1 \rangle, \langle \texttt{x}, \delta_2 \rangle, \phi_1, \ldots, \phi_m$$

This definition of **assume** allows us to simply use **move** to deal with hypothesis manipulation, which then is subject to the same constraints as normally moving objects:

$$\textbf{move3}(\langle \nu_1, \texttt{+y}\delta_1 \rangle, \phi_1, \ldots, \langle \delta, \texttt{-y}\delta_2 \rangle, \ldots, \phi_m) = \langle \nu_1, \delta_1 \rangle, \phi_1, \ldots, \langle \delta\texttt{-y}, \delta_2 \rangle, \ldots, \phi_m$$

Once all but one of the features of a hypothesis have been 'checked', it is ready to be discharged. As with **merge**, the definition of **discharge** is split into two cases, as determined by whether the second argument will continue moving (i.e. whether it has licensee features in need of checking).

$$\textbf{discharge1}(\langle \nu_1, \texttt{+y}\delta_1 \rangle, \phi_1, \ldots, \langle \delta, \texttt{-y} \rangle, \ldots, \phi_m; \langle \nu_2, \delta\texttt{-y} \rangle, \psi_1, \ldots, \psi_n) = \langle \nu_2\nu_1, \delta_1 \rangle, \phi_1, \ldots, \phi_m, \psi_1, \ldots, \psi_n$$

$$\textbf{discharge2}(\langle \nu_1, \texttt{+y}\delta_1 \rangle, \phi_1, \ldots, \langle \delta, \texttt{-y} \rangle, \ldots, \phi_m; \langle \nu_2, \delta\texttt{-y}\delta_2 \rangle, \psi_1, \ldots, \psi_n) = \langle \nu_1, \delta_1 \rangle, \phi_1, \ldots, \phi_m, \langle \nu_2, \delta_2 \rangle, \psi_1, \ldots, \psi_n$$

As with **move**, we require that the arguments to **discharge** satisfy the SMC.

discharge$(\langle \nu, \texttt{+y}\delta \rangle, \phi_1, \ldots, \phi_m; \Psi)$ is defined only if at most one $\phi_i = \langle \alpha, \delta_i \rangle$ is such that δ_i begins with `-y`

[5] Note that **assume** is a relation. I don't see any obvious way to incorporate hypothetical reasoning into the minimalist grammar framework without some element of non-determinism. The intuitive presentation as given in figure 4, where each application of **assume** hypothesizes away just a single feature, moves the non-determinism into the bookkeeping for which hypotheses may be eliminated by a single instance of the **discharge** operation. (Consider how many dotted lines we could have drawn in figure 5 if there had been other $\texttt{t}_\texttt{d}$ hypotheses in the first argument to **discharge**.) Here, I have opted to localize all of the non-determinism in the **assume** operation, in the hope that this will make it easier to optimize away. (An obvious optimization is to limit the choice of δ_2 to only those sequences of licensee features that actually occur in the lexicon.) It certainly is easier to present this way.

Some comments on the formalism. Before moving on to some examples of the formalism at work, it is worth pointing out the following. First, it is clear that extending minimalist grammars with the operations **assume** and **discharge** in the manner described above does not increase the weak generative capacity of the formalism (the proof is a straightforward modification of the embedding given in [19] of minimalist grammars in MCFGs). Second, the particular definitions of **assume** and **discharge** given here allow for a certain kind of 'smuggling' [10] of moving subexpressions (to be taken up in §2.3). Specifically, certain violations of the SMC can be gotten around by delaying the first merge of an expression containing subexpressions which otherwise would compete with subexpressions of the main expression for checking. The smuggling seems logically independent of the addition of hypothetical reasoning to the minimalist grammar system [1], although it is not immediately obvious how to give a similarly elegant hybrid system without it.

2.3 An Analysis in the Hybrid Framework

In this section I will illustrate the workings of the hybrid framework developed in §2.2 by couching an analysis of passivization and relativization in English in these terms. Passivization is a canonical example of A movement, and relativization of A-bar movement. For relativization, I take as my starting point the raising analysis reanimated recently by [14] (see also [2,3,15,30]), according to which the head noun modified by the relative clause is base generated within the relative clause modifying it, and raised to its surface position. Schematically, one has derivations like the following.

man [that the book will be written by `t`]

For the analysis of passive, I adopt the smuggling analysis advanced recently by [10], according to which the demoted subject in the *by*-phrase is base generated in its canonical position. Under Collins' analysis, a passive sentence is derived by moving the participle phrase to the specifier of the passive voice phrase (which is headed by *by*), and then exceptionally moving the logical object out from inside the just moved participle phrase into the surface subject position, as schematized below.

[the book] will be [written `t`] by the man `t`

Determiner phrases will be assigned the type '`d -k`', which means that they will surface in a different position (`-k`) than the one they were merged in (`d`). This is an implementation of the idea that DPs move for case [8]. There are two 'determiners' in our fragment:

i. $\langle$`the`, `=n d -k`$\rangle$
ii. $\langle\epsilon$, `=n d -k -rel`$\rangle$

The first is the familiar *the*, the typing of which lets us know that it selects a noun phrase (of type `n`), and is then a determiner phrase (of type `d -k`). The second is particular to the raising analysis of relative clauses. It allows a noun phrase to behave as a determiner phrase within a clause, and then raise out of the clause (-rel), forming a noun-relative clause compound.

iii. ⟨that, =s +rel n⟩

Lexical item iii selects a clause (of type s), and triggers raising of a noun phrase (+rel). The result (N Rel) behaves as a noun phrase (n).

iv. ⟨smile, =d v⟩
v. ⟨will, =v +k s⟩
vi. ⟨man, n⟩

With the addition of lexical items iv, v, and vi, we may derive sentences like *the man will smile* in the following manner.

1. **assume**(iv)
   ⟨smile, v⟩, ⟨d, -k⟩
2. **merge**(v,1)
   ⟨will smile, +k s⟩, ⟨d, -k⟩
3. **merge**(i,vi)
   ⟨the man, d -k⟩
4. **discharge**(2,3)
   ⟨the man will smile, s⟩

The relative clause *man that will smile* has an initially similar derivation, first diverging at step 3:

3. **merge**(ii,vi)
   ⟨man, d -k -rel⟩
4. **discharge**(2,3)
   ⟨will smile, s⟩, ⟨man, -rel⟩
5. **merge**(iii,4)
   ⟨that will smile, +rel n⟩, ⟨man, -rel⟩
6. **move**(5)
   ⟨man that will smile, n⟩

With lexical items i–vi, all and only sentences belonging to the regular set the man (that will smile)* will smile are derivable. Expanding our fragment, we turn next to transitive clauses in the active voice, for which we need the following new lexical items.[6]

vii. ⟨write, =d V⟩
viii. ⟨ϵ, =>V +k V⟩

[6] The feature type =>x is a variant of =x, one which in addition triggers movement of the selected phrase's head. For details, see [28,20,15].

ix. $\langle\epsilon,$ `=>V =d v`$\rangle$
x. $\langle$book, `n`$\rangle$

Lexical item viii allows the object to check its case (`-k`) within the extended projection of the verb (the head movement is to get the word order right). It is optional, as passivization requires the object to check its case *outside* of the verb phrase in English. Lexical item ix is the head which selects the external argument of the verb phrase, and changes the category of the verbal projection to 'little v' (`v`). The sentence *the man will write the book* has the following derivation.

1. **assume**(vii)
   ⟨write, `V`⟩, ⟨`d`, `-k`⟩
2. **merge**(viii,1)
   ⟨write, `+k V`⟩, ⟨`d`, `-k`⟩
3. **merge**(i,x)
   ⟨the book, `d -k`⟩
4. **discharge**(2,3)
   ⟨the book write, `V`⟩
5. **merge**(ix,4)
   ⟨write the book, `=d v`⟩
6. **assume**(5)
   ⟨write the book, `v`⟩, ⟨`d`, `-k`⟩
7. **merge**(v,6)
   ⟨will write the book, `+k s`⟩, ⟨`d`, `-k`⟩
8. **merge**(i,vi)
   ⟨the man, `d -k`⟩
9. **discharge**(7,8)
   ⟨the man will write the book, `s`⟩

To derive passive sentences, we require the following five lexical items.

xi. ⟨-en, `=>V V -part`⟩
xii. $\langle\epsilon,$ `=v +k x`$\rangle$
xiii. ⟨by, `=x +part pass`⟩
xiv. $\langle\epsilon,$ `=v +part pass`$\rangle$
xv. ⟨be, `=pass v`⟩

Using these lexical items, we may derive the sentence *The book will be written by the man* in the following manner.

1. **assume**(ix)
   $\langle\epsilon,$ `=d v`$\rangle$, ⟨`V`, `-part`⟩
2. **assume**(1)
   $\langle\epsilon,$ `v`$\rangle$, ⟨`d`, `-k`⟩, ⟨`V`, `-part`⟩

3. **merge**(xii,2)

$$\langle \epsilon, \texttt{+k x} \rangle, \langle \texttt{d}, \texttt{-k} \rangle, \langle \texttt{V}, \texttt{-part} \rangle$$

4. **merge**(i,vi)

$$\langle \textsf{the man}, \texttt{d -k} \rangle$$

5. **discharge**(3,4)

$$\langle \textsf{the man}, \texttt{x} \rangle, \langle \texttt{V}, \texttt{-part} \rangle$$

6. **merge**(xiii,5)

$$\langle \textsf{by the man}, \texttt{+part pass} \rangle, \langle \texttt{V}, \texttt{-part} \rangle$$

7. **assume**(vii)

$$\langle \textsf{write}, \texttt{V} \rangle, \langle \texttt{d}, \texttt{-k} \rangle$$

8. **merge**(xi,7)

$$\langle \textsf{written}, \texttt{V -part} \rangle, \langle \texttt{d}, \texttt{-k} \rangle$$

9. **discharge**(6,8)

$$\langle \textsf{written by the man}, \texttt{pass} \rangle, \langle \texttt{d}, \texttt{-k} \rangle$$

10. **merge**(xv,9)

$$\langle \textsf{be written by the man}, \texttt{v} \rangle, \langle \texttt{d}, \texttt{-k} \rangle$$

11. **merge**(v,10)

$$\langle \textsf{will be written by the man}, \texttt{+k s} \rangle, \langle \texttt{d}, \texttt{-k} \rangle$$

12. **merge**(i,x)

$$\langle \textsf{the book}, \texttt{d -k} \rangle$$

13. **discharge**(11,12)

$$\langle \textsf{the book will be written by the man}, \texttt{s} \rangle$$

Note that if expression 8 were merged directly in step 1 instead of having been assumed in 1 and then discharged in 9, the derivation would have crashed at step 5, as *the man* and (the hypothesis of) *the book* would have been competing for the same position. We may derive the relative clause *book that will be written by the man* by first recycling steps 1–11 of the previous derivation, and then continuing in the following manner.

12. **merge**(ii,x)

$$\langle \textsf{book}, \texttt{d -k -rel} \rangle$$

13. **discharge**(11,12)

$$\langle \textsf{will be written by the man}, \texttt{s} \rangle, \langle \textsf{book}, \texttt{-rel} \rangle$$

14. **merge**(iii,13)

$$\langle \textsf{that will be written by the man}, \texttt{+rel n} \rangle, \langle \textsf{book}, \texttt{-rel} \rangle$$

15. **move**(14)

$$\langle \textsf{book that will be written by the man}, \texttt{n} \rangle$$

3 Conclusions

I have demonstrated how a straightforward modification of the minimalist grammar framework yields a formal architecture for the description of language in which there exist two kinds of movement dependencies, which obey the ban on improper movement. It is simple and natural to connect the structure of movement chains to semantic interpretation in a way that derives the crossover differences between A and A-bar movement.[7] Further semantic asymmetries, such as the hypothesis that A movement systematically prohibits reconstruction while A-bar movement does not [9,16], are also easily incorporable.[8] One major difference between the hybrid minimalist framework presented here and the intuitive conception of A and A-bar movement present in the GB/MP literature, is the lack of a decision procedure for determining *when* an expression stops A moving and starts A-bar moving. (This is related to the fact that this decision procedure in the GB/MP relies on a complex network of assumptions which in the MG framework are non-logical, such as universal clausal structure, and cross-linguistic identity of features.) As mentioned in footnote 8, this freedom allows us to pursue hypotheses about language structure that we otherwise could not. It remains to be seen whether these novel hypothesis types prove enlightening, as it does the broader A/A-bar distinction.

References

1. Amblard, M.: Calculs de représentations sémantiques et syntaxe générative: les grammaires minimalistes catégorielles. Ph.D. thesis, Université Bordeaux I (2007)
2. Bhatt, R.: The raising analysis of relative clauses: Evidence from adjectival modification. Natural Language Semantics 10(1), 43–90 (2002)
3. Bianchi, V.: The raising analysis of relative clauses: A reply to Borsley. Linguistic Inquiry 31(1), 123–140 (2000)
4. Boeckx, C.A.: A note on contraction. Linguistic Inquiry 31(2), 357–366 (2000)
5. Bresnan, J.: Lexical-Functional Syntax. Blackwell, Oxford (2001)
6. Büring, D.: Crossover situations. Natural Language Semantics 12(1), 23–62 (2004)
7. Chomsky, N.: On Wh-movement. In: Culicover, P.W., Wasow, T., Akmajian, A. (eds.) Formal Syntax, pp. 71–132. Academic Press, New York (1977)
8. Chomsky, N.: Lectures on Government and Binding. Foris, Dordrecht (1981)
9. Chomsky, N.: The Minimalist Program. MIT Press, Cambridge (1995)

[7] An expression may bind all and only those expressions it c-commands in the derivation tree. Note that this is a formal rendering of what [6] calls 'Reinhart's Generalization' (after [24]):

Pronoun binding can only take place from a c-commanding A position.

[8] Two possibilities suggest themselves. First, we might allow an expression to reconstruct into any position through which it has moved (i.e. between where it is first merged/discharged and where it is last moved) [16]. Another option is to abandon the idea that there are dedicated A and A-bar positions, and force an expression to be interpreted exactly where it is first merged/discharged.

10. Collins, C.: A smuggling approach to the passive in English. Syntax 8(2), 81–120 (2005)
11. Fox, D.: Reconstruction, binding theory, and the interpretation of chains. Linguistic Inquiry 30(2), 157–196 (1999)
12. Gärtner, H.M., Michaelis, J.: A note on the complexity of constraint interaction: Locality conditions and minimalist grammars. In: Blache, P., Stabler, E.P., Busquets, J.V., Moot, R. (eds.) LACL 2005. LNCS (LNAI), vol. 3492, pp. 114–130. Springer, Heidelberg (2005)
13. Gazdar, G.: Unbounded dependencies and coordinate structure. Linguistic Inquiry 12(2), 155–184 (1981)
14. Kayne, R.: The Antisymmetry of Syntax. MIT Press, Cambridge (1994)
15. Kobele, G.M.: Generating Copies: An investigation into structural identity in language and grammar. Ph.D. thesis, University of California, Los Angeles (2006)
16. Lasnik, H.: Chains of arguments. In: Epstein, S.D., Hornstein, N. (eds.) Working Minimalism in Current Studies in Linguistics, vol. 32, pp. 189–215. MIT Press, Cambridge (1999)
17. Lasnik, H., Stowell, T.: Weakest crossover. Linguistic Inquiry 22, 687–720 (1991)
18. Mahajan, A.: The A/A-bar Distinction and Movement Theory. Ph.D. thesis, Massachusetts Institute of Technology (1990)
19. Michaelis, J.: On Formal Properties of Minimalist Grammars. Ph.D. thesis, Universität Potsdam (2001)
20. Michaelis, J.: Notes on the complexity of complex heads in a minimalist grammar. In: Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6), Venezia (2002)
21. Müller, G., Sternefeld, W.: Improper movement and unambiguous binding. Linguistic Inquiry 24(3), 461–507 (1993)
22. Pollard, C.J., Sag, I.A.: Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago (1994)
23. Postal, P.M.: Cross-over phenomena. Holt, Rinehart & Winston, New York (1971)
24. Reinhart, T.: Anaphora and Semantic Interpretation. University of Chicago Press, Chicago (1983)
25. Ross, J.R.: Constraints on Variables in Syntax. Ph.D. thesis, Massachusetts Institute of Technology (1967), published in 1986, as Infinite Syntax, Ablex
26. Ruys, E.G.: Weak crossover as a scope phenomenon. Linguistic Inquiry 31(3), 513–539 (2000)
27. Sportiche, D.: Reconstruction, binding and scope, ms. UCLA (2005)
28. Stabler, E.P.: Derivational minimalism. In: Retoré, C. (ed.) LACL 1996. LNCS (LNAI), vol. 1328, pp. 68–95. Springer, Heidelberg (1997)
29. Stabler, E.P., Keenan, E.L.: Structural similarity within and among languages. Theoretical Computer Science 293, 345–363 (2003)
30. de Vries, M.: The Syntax of Relativization. Ph.D. thesis, Universiteit van Amsterdam (2002)