The Configurational Matrix

Ad Neeleman
Hans van de Koot

This article presents a theory of grammatical dependencies that is in accordance with basic assumptions of bare phrase structure theory. It explains Koster’s (1987) configurational matrix, the observation that such dependencies share five properties: c-command by the antecedent, obligatoriness, uniqueness of the antecedent, nonuniqueness of the dependent, and locality. The theory is based on two primitive syntactic relations (copying and function application) and a nonatomic view of nodes.

Keywords: Inclusiveness, phrase structure, dependencies, trace theory, theta theory

1 Dependencies and Inclusiveness

Bare phrase structure theory (Chomsky 1995a) reduces the machinery involved in the generation of syntactic trees to a minimum. Its basic assumption, Inclusiveness, states that properties of syntactic structure must derive from information stored in the lexicon. As a result, bar levels, coindexation, and so on, must be dispensed with.

Chomsky’s (1995b:228) discussion suggests that Inclusiveness is intended to be uniform: it holds not only of the root node of a sentence, but also of every subtree contained in it. This property is highlighted by (1) (where ‘‘mapping procedures’’ link a terminal to a lexical entry).1

(1) Inclusiveness
The syntactic properties of a nonterminal node are fully recoverable from the structure it dominates; the syntactic properties of a terminal node are fully recoverable through mapping procedures.

The standard analysis of grammatical dependencies is at odds with Inclusiveness. Binding, movement, predication, and the licensing of negative polarity items (NPIs) are typically thought of as involving a chainlike relation between a dependent and its c-commanding antecedent. This relation often affects properties of the antecedent or dependent. Predication, for example, has the consequence that one of the predicate’s θ-roles is satisfied, while the argument cannot satisfy any of

We thank the audiences at the Université Sidi Mohamed Ben Abdellah, at the IVth International LEHIA Workshop (University of the Basque Country), and at Tools in Linguistic Theory 1 (Utrecht University). We also thank Klaus Abels, Peter Ackema, Michael Brody, Ulf Brosziewski, Dirk Bury, Annabel Cormack, Frank Drijkoningen, John Harris, Richard Hudson, Marcus Kracht, Marika Lekakou, Eric Mathieu, Alex Perović, Eric Reuland, Vieri Samek-Lodovici, Neil Smith, Ed Stabler, Kriszta Szendrői, Vina Tsakali, Reiko Vermeulen, Fred Weerman, Edwin Williams, and two LI reviewers.

1 Nonuniform Inclusiveness would allow scattering of the features of a lexical item across terminals, sideways transfer of features, and introduction of features in nonterminals.
the predicate’s other roles. Thus, both the argument and the predicate acquire new properties. However, these cannot be recovered from their internal structures; any subtree that contains the predicate but not its argument, or vice versa, is noninclusive in the sense of (1).

So, a condition central to phrase structure does not seem to hold of grammatical dependencies. There are two possible reactions to this problem. The first is to deny that dependencies are encoded by the syntax; they would be established syntax-externally instead. This is the line taken by Brody (1998b) for movement chains. The second is to develop a new encoding of dependencies that does not violate Inclusiveness. An argument for the second option can be based on Koster’s (1987) configurational matrix.

Koster observes that grammatical dependencies share five properties:

1. The dependent must take an antecedent.
2. The antecedent must be in a c-commanding position.
3. The antecedent must be sufficiently close to the dependent.
4. Each dependent must take a unique antecedent.
5. An antecedent can be linked to multiple dependents.

This configurational matrix must be inherent in the way syntax operates, not only because it refers to syntactic configurations, but also because it does not hold of syntax-external phenomena such as logophoric relations (Reinhart and Reuland 1991, 1993), coreference (Reinhart 1983, 1996), and bound variable anaphora (Reuland 2001).2 We will not discuss control, but differences between obligatory and nonobligatory control also demonstrate the contrast between syntactic and extra-syntactic relations (Williams 1980).

The configurational matrix provides an argument not only for the syntactic encoding of grammatical dependencies but also for revising the way in which this is done. Standard approaches do not explain why relations prima facie very different in nature display a constant set of properties. Government-Binding (GB) Theory, for example, simply imposes conditions on the various rules that associate dependents with their antecedents. Such an approach is unparsimonious and arbitrary: why should such rules all be conditioned in the same way, and why should they meet the conditions in question rather than others?

Progress can be made if some grammatical dependencies are reduced to others. Chomsky (1986b), for instance, reanalyses binding as LF movement, while Progovac (1994) suggests that NPI licensing is a subcase of binding. Similarly, Manzini and Roussou (2000) claim that θ-role assignment is movement of an aspectual feature. This trend is foreshadowed in Koster 1987, where a metarelation R subsumes all dependencies.

But even if parsimoniousness is thus guaranteed, it remains unclear why the metarelation R is conditioned in the way it is. In addition, reductionism cannot easily deal with properties that distinguish grammatical dependencies. Movement, for example, obeys the Adjunct Condition but

---

2 Bound variable anaphora is sometimes argued to exhibit a c-command condition (Ruys 2000). Reuland (2001) attributes this to compositionality rather than syntactic encoding.
binding does not. Similarly, predication and movement are mutually exclusive in that they cannot share an antecedent (there is no raising to $\theta$-positions), whereas shared antecedents are allowed in the case of predication and binding. If all grammatical dependencies were one, these contrasts would be hard to capture.

This article presents an alternative theory of grammatical dependencies based on four assumptions. The first three are Economy, Inclusiveness, as formulated in (1), and Accessibility, according to which relations between nodes require immediate domination. These assumptions, we argue, imply that dependencies must be decomposed into more primitive relations. More specifically, a dependency is established if a selectional function introduced by a dependent is copied upward until it directly dominates a node that satisfies it. This approach provides an account of why c-command and obligatoriness are part of the configurational matrix. Our fourth assumption, Distinctness, is that attributes of nodes (features and selectional functions) must be distinguished either by their inherent properties or by ordering. Distinctness explains why antecedents are unique, whereas dependents need not be. Our four assumptions also underlie aspects of locality, the final property of the configurational matrix, and various other phenomena.

2 Grammatical Dependencies

2.1 Function Application

The fact that a grammatical dependent requires an antecedent must be encoded as one of its syntactic properties. There are several ways in which this can be done, and which one is adopted does not affect our argumentation. For concreteness’ sake, we conceptualize selectional requirements as functions satisfied by particular features. A verb that selects a prepositional complement contains a function $P$, which looks for an appropriate set of categorial features. $P$ can be formalized as the identity function, with its domain specified as $P$. In the same vein, predicates, NPIs, anaphors, and traces introduce an identity function but in each case with the function’s domain specified differently.

We do not intend the functions we introduce to be semantic in nature. They are syntactic objects used to establish syntactic dependencies. Such dependencies often receive a semantic interpretation at LF, but this will not concern us. Furthermore, as in other theories, the question arises which grammatical dependencies must be distinguished. We largely ignore this question, although we make some specific proposals concerning the individuation of functions mediating movement.

We will now discuss which structural description holds of function application: sisterhood, containment, or direct domination.

2.2 Function Application under Sisterhood

The formation of complex categories is often taken to require upward copying of lexical information. A related view is that grammatical dependencies are established under zero, one, or more steps of upward copying of a function, followed by application under sisterhood. Consider (2). If the dependent $\alpha$ is related to the antecedent $\delta$, $\alpha$ introduces a function $\mathcal{F}$ that is copied upward
to $\gamma$, where it is satisfied by $\delta$ under sisterhood (‘#’ indicates that $\mathcal{F}$ has been satisfied). This analysis presupposes the rule in (3).

\[
(2) \quad \varepsilon
\]

\[
\delta \quad \gamma \quad [\mathcal{F}_#]
\]

\[
\alpha \quad [\mathcal{F}] \quad \beta
\]

(3) **Function application**

$\mathcal{F}$ is read as satisfied by $\alpha$ if and only if $\text{Contain} (n_1, \mathcal{F}) \& \text{Contain} (n_2, \alpha) \& \text{Sisters} (n_1, n_2)$.

Although widely assumed, function application under sisterhood is not unproblematic.

First, sisterhood is a sideward relation. If the function in (2) is satisfied by $\delta$ under sisterhood, then $\gamma$ acquires properties that cannot be recovered from the structure it dominates (its daughters $\alpha$ and $\beta$). The subtree rooted in $\gamma$ hence violates Inclusiveness.

Second, nodes in a sideward relation have a common ancestor. In (2) sisterhood between $\delta$ and $\gamma$ holds in virtue of $\varepsilon$. The usual definition of sisterhood in terms of direct domination by the same node is unexpected if sisterhood is a primitive, as (3) suggests. Put differently, since indispensable relations between $\delta$ and $\varepsilon$ and between $\gamma$ and $\varepsilon$ already connect $\delta$ and $\gamma$, an independent—sideward—association of $\delta$ and $\gamma$ seems redundant. Put still differently, function application under sisterhood does not adhere to the null hypothesis in (4).

(4) **Accessibility**

Relations between nodes require immediate domination.

A theory not based on Accessibility allows relations that do not have a ‘‘medium,’’ so that any node in a tree can in principle be directly related to any other. Hence, in addition to sisterhood, other relations can be formulated, many of which are not attested.

Third, function application under sisterhood gives rise to an empirical problem. Any theory must distinguish between internal and external dependencies. Anaphoric binding and NPI licensing involve dependencies between a maximal projection and an element external to it, while direct $\theta$-marking and c-selection hold between a head and an element contained in its projection. We now show that the locality of internal dependencies does not follow in theories based on upward copying and function application under sisterhood.

The projection of a head consists of those nodes to which its categorial features are copied. This allows a natural distinction between internal and external dependencies. The former involve functions that can be copied as far as the head’s categorial features. The latter involve functions $\varepsilon$.

---

3 See Koster 2000, where the metarelation $\mathcal{R}$ of Koster 1987 is restricted to sisterhood.
that can be copied independently of categorial features and hence beyond the maximal projection of the head that introduces them.

If function application under sisterhood is assumed, internal dependencies cannot be kept internal. This can be demonstrated on the basis of c-selection. Although a head may select the category of arguments within its maximal projection, an argument may not select for the category of the head in whose projection it is contained. Thus, a verb may select a prepositional complement (see (5a)) but a prepositional complement may not select a verb (see (5b)). However, if c-selection is constrained by sisterhood, it is hard to understand why the category of the node dominating V and PP should matter.

(5) a. V [P #] b. * V

V [P #] P V [P [γ#]]

There are ways in which this problem can be circumvented, but since these must all refer to the dominating node, sisterhood cannot be the correct structural description for function application. Moreover, we have shown that function application under sisterhood is not compatible with Inclusiveness and that sisterhood is an unlikely primitive of grammar, as it is itself defined in terms of domination.

2.3 Function Application under Containment

The situation can be improved if sisterhood is decomposed into relations that hold independently between each sister and its mother. Suppose, for instance, that upward copying is the only relation that can hold between nodes. The structural description of function application compatible with this is containment: a function is satisfied if copied to the same node as the feature that forms its argument.

(6) Function application (revised)

\( P \) is read as satisfied by \( \alpha \) if and only if \( \text{Contain} (n, P) \& \text{Contain} (n, \alpha) \).

C-selection may again serve to illustrate the approach. If a verb c-selects a prepositional complement, it introduces a function satisfied by particular categorial features. This function is copied from the verb to the dominating node, as are the categorial features of the prepositional complement. After copying, the function is satisfied under containment, as in (7).

(7) V [P, [−V, −N]]

V [P] P [−V, −N]

One advantage of (6) is that it keeps internal dependencies internal. If a c-selectional function cannot be copied out of the maximal projection of the head that introduces it, its argument features must originate in a constituent within this projection, as desired.
However, c-selection also highlights three problems with function application under containment. First, argument features may clash with those of the node that hosts the function they satisfy. In (7), for instance, the root node contains both verbal and prepositional categorial features. These are normally incompatible and therefore a statement is required that assigns the argument features a special status.

Second, it is not obvious that argument features can always be copied out of the constituent in which they originate. This is particularly clear where categorial features are concerned. Since such features are copied up to a head’s maximal projection but not beyond it, the copy operation from P to V in (7) is hard to motivate.

Third, there are heads that c-select a category of the same type, such as determiners selecting an NP (Hudson 1987, Abney 1987, Doetjes, Neeleman, and Van de Koot 1999). Such heads contain the features that the function they introduce looks for. Given (6), the conditions for function application are met in the head and therefore the function should count as satisfied there. To avoid this problem, the origin of features must be taken into account: the features that form the argument of a function must be copied from a node that does not contain the function itself. This is expressed by (8).

\[
\text{(8) Function application (revised)}
\]

\[
\mathcal{F} \text{ is read as satisfied by } \alpha \text{ if and only if } \text{Contain} (n_1, \mathcal{F}) \land \text{Contain} (n_1, \alpha) \land \text{Dominate} (n_1, n_2) \land \text{Contain} (n_2, \alpha) \land \lnot \text{Contain} (n_2, \mathcal{F}).
\]

This reformulation of (6) is suspicious. Since a function must be satisfied by information that can be traced back to a node different from the one that hosts the function, why should the relevant information be copied to the node hosting that function in the first place?

### 2.4 Function Application under Direct Domination

This brings us to our own proposal. As we have shown, function application cannot be a relation between sisters and it cannot require containment either. This leaves just one option: like copying, it should be conditioned by direct domination. But whereas copying is an upward relation (a function is copied to a dominating node), function application is downward (a function is read as satisfied if it directly dominates its argument). That is, instead of (8), we adopt (9), in which the redundant clause “\(\text{Contain} (n_1, \alpha)\)” is omitted.

\[
\text{(9) Function application (revised)}
\]

\[
\mathcal{F} \text{ is read as satisfied by } \alpha \text{ if and only if } \text{Contain} (n_1, \mathcal{F}) \land \text{Dominate} (n_1, n_2) \land \text{Contain} (n_2, \alpha) \land \lnot \text{Contain} (n_2, \mathcal{F}).
\]

C-selection can now be analyzed as follows. Through upward copying, a c-selectional function is transferred from the head that introduces it to the node directly dominating the complement, after which the downward relation between mother and daughter allows for satisfaction. In other words, a c-selectional function is satisfied if it directly dominates a node of the correct syntactic category, as in (10).
On this approach internal dependencies remain internal (assuming, as before, that internal functions cannot be copied out of maximal projections).

There is a crucial difference between the upward and downward relations in (10). The former is one by which information is repeated in a higher node, whereas the latter uses a lower node to establish the status of a function. This is not a quirk of c-selection but reflects the way Inclusiveness regulates the flow of information in a tree: upward relations must be based on copying and downward relations on function application.

Consider why downward copying violates Inclusiveness. Suppose a function originating in a head X is copied upward onto X’s maximal projection and then copied downward onto some other maximal projection, as in (11).

The structure in (11) is ruled out, as Y’s maximal projection contains a function not present in either of its daughters. Consequently, this node is partially noninclusive. A similar line of argumentation applies to downward copying to terminals. We conclude that Inclusiveness rules out downward copying.

Next, consider why upward function application violates Inclusiveness. Such a relation would amount to a node imposing a requirement on its mother, a type of selection not attested. Let us briefly return to the illicit selection of a verb by a prepositional complement. If upward application were allowed, such selection could be encoded through a function $\gamma$, present on the PP and satisfied by the dominating VP, as in (12).

Inclusiveness requires that the properties of a node be recoverable from its daughters. This is not true of the PP in (12). The function on this node is satisfied, but this is not due to one of its daughters. Hence, upward function application, like downward copying, violates Inclusiveness.

The above assumes that grammatical dependencies are established by function application. This need not be stipulated. Although features as well as functions can be copied, it is impossible to establish a dependency by transferring a feature from dependent to antecedent or vice versa.
Such transfer must involve a step of downward copying, in violation of Inclusiveness. Thus, function application is the only viable way of establishing grammatical relations.

The ramifications of the above conclusions are explored in section 3. However, we first consider other theories based on copying and function application.

2.5 Function Application in Other Frameworks

Function application is particularly prominent in Categorial Grammar (CG; Steedman 1993, 2000b, and references mentioned there). However, since CG does not assume phrase structure, it is impossible to speak of the structural configuration under which function application takes place. Rather, as in standard lambda calculus, a function is applied to its argument under juxtaposition, a linear notion. This implies that the results we derive from Accessibility and Inclusiveness do not carry over to CG.

Our proposals are closer to unification-based approaches such as Lexical-Functional Grammar (LFG; Kaplan and Bresnan 1982, Bresnan 2000), Generalized Phrase Structure Grammar (GPSG; Gazdar et al. 1985), and Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag 1987, 1994). These are based on feature sharing, a notion similar to copying, and the satisfaction of lexical constraints through unification, a notion similar to function application. Consider, for instance, the slash features used in GPSG and HPSG to encode movement. Such features are introduced by a dependent element, copied upward and satisfied by the antecedent.

In LFG, GPSG, and HPSG the standard environment for resolution is sisterhood—as is clear from HPSG’s Subcategorization Principle, for example. We have argued in section 2.2 that dependencies established under sisterhood are at odds with Accessibility and Inclusiveness. Nevertheless, it would be incorrect to infer an incompatibility between the ideas developed here and unification-based theories. Containment and domination are possible alternatives for sisterhood in any theory that assumes trees. (13a) represents standard resolution under sisterhood, while (13b) and (13c) correspond to resolution under containment and domination, respectively. In these structures $\alpha$ contains the category that introduces a slash feature /F and $\beta$ contains F, the feature that satisfies it. So, an implementation of our proposals in unification-based theories is feasible.

\[
\begin{align*}
(13) & \quad (a) \quad \gamma \quad \beta [F] \\
& \quad \alpha/F_# \\
(b) & \quad \gamma/F_# [F] \\
& \quad \alpha/F \quad \beta [F] \\
(c) & \quad \gamma/F_# \\
& \quad \alpha/F \quad \beta [F]
\end{align*}
\]

3 C-Command

3.1 Deriving C-Command

Grammatical dependencies require that the dependent element be c-commanded by its antecedent. A trace, for instance, must be c-commanded by the moved element, an anaphor by its binder,
and a predicate by its subject. In standard analyses dependencies are “direct”: they are established without transmission of information through intervening nodes. Why a direct relation should require c-command is unclear.

By contrast, the combination of Accessibility and Inclusiveness does explain why c-command restricts grammatical dependencies. A grammatical dependency must involve a function introduced by a dependent. Inclusiveness forces copying of this function to be upward and application to be downward. Since copying involves transfer of information, it can apply recursively: a function copied to a dominating node can subsequently be copied from that node to its mother, and so on. The downward relation of function application is strictly bounded, however. A function can be satisfied only by a daughter of the node that hosts it. Satisfaction by a lower node would violate Accessibility.

The implication is that the element satisfying the function on which a dependency is based does not have to be a sister of the element introducing it. The function $F$ in (14a) can be satisfied by $\beta$, or if $\beta$ is not of the right type, the function can be copied onto $\epsilon$ and applied to $\delta$. The situation is different in (14b). $F$, once copied onto $\epsilon$, can be satisfied by $\gamma$. Given Accessibility, it cannot be applied to either $\beta$ or $\delta$, however.

Thus, the first property of the configurational matrix is derived: if $\alpha$ is grammatically dependent on $\beta$, $\beta$ c-commands $\alpha$. This result follows from the decomposition of grammatical dependencies into upward copying and downward function application, which is itself necessary to reconcile such dependencies with Inclusiveness.

3.2 Effects for Grammatical Dependencies

Let us briefly illustrate the effects of the above proposals for c-selection. Here the dependent is the verb: it introduces a function $Cal$ that is satisfied by the selected category (the function $P$.

4 This claim has been challenged. Branching dependencies, such as parasitic gaps, present a problem for derivational approaches to syntax, since two moved elements must be merged in a single position. Nunes (2001) allows sideward movement, which makes it possible to maintain that there is a single moved element in branching dependencies. Crucially, however, only movement to a c-commanding position involves chain formation. Therefore, c-command as a property of movement chains is not disputed.

5 Epstein (1995) relates the c-command condition to a derivational view of syntax. A detailed critique of this analysis is offered by Brody (1998a). Brody’s main point is that merger, an essentially symmetrical operation, cannot underlie c-command restrictions, which are essentially asymmetrical.
mentioned earlier is an instantiation). The selected category can thus be seen as the antecedent: it caters to an unsaturated property of the verb. Hence, the selected category must c-command the verb. In (15a) \textit{Cat} can be satisfied by \(\alpha\), but not by \(\beta\) or \(\gamma\). Indeed, a verb cannot c-select the specifier or complement of its complement. In contrast, the verb need not c-command the selected element. In (15b) \textit{Cat} can be satisfied by \(\alpha\) or, if this is not the case and the function is copied onto VP, by \(\beta\).

\[
\begin{align*}
\text{(15) a.} & \quad \begin{array}{c}
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\end{array} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\text{V [Cat]} \\
\end{align*}
\]

Again, this is correct: in (16a) the PP selected by \textit{looked} is a sister of the verb; in (16b) it is separated from the verb by an adverbial.\footnote{The same point can be made if (16a–b) contain a VP shell, whose specifier is filled by \textit{at the telegram} in (16a) and by \textit{pensively} in (16b) (Chomsky 1995b). On this view the verb c-selects for a nonsister in (16a).}

\[
\begin{align*}
\text{(16) a.} & \quad \begin{array}{c}
\text{The director [looked at the telegram] pensively].} \\
\text{The director [looked pensively] at the telegram].} \\
\end{array} \\
\end{align*}
\]

We expect this pattern to extend to other grammatical dependencies, given that these must also be encoded by functions. For movement this implies that the trace introduces a function comparable to the slash feature in GPSG and HPSG. For now we refer to this function as \textit{Move}; later on we will have reason to distinguish functions for A-, \(\bar{A}\)-, and head movement. A-movement is illustrated in (17).

\[
\begin{align*}
\text{(17) a.} & \quad \begin{array}{c}
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\text{\[Move\]} \\
\end{array} \\
\end{align*}
\]

If movement relies on function application, it is correctly predicted that there can be no downward or sideward movement. These restrictions are illustrated by the contrasts in (18a–b) and (18c–d), respectively. (For head adjunction, see section 10.)

\[
\begin{align*}
\text{(18) a.} & \quad \begin{array}{c}
\text{John explained [that [to children] one should not tell such stories} \ t_{pp}].} \\
\text{*John explained} \ t_{pp} \ [\text{that} \ t_{pp} \ \text{to Mary] safe is better than sorry].} \\
\text{c.} \quad \begin{array}{c}
\text{[Deciding [which movie to see} \ t_{wh} \ \text{next]] makes John very happy.} \\
\text{d.} \quad \begin{array}{c}
\text{*[Deciding [who to see that new movie next]] makes} \ t_{wh} \ \text{very happy.} \\
\end{array} \\
\end{array} \\
\end{align*}
\]
Anaphors, like traces, are dependent elements. They introduce a function, which we call \( \text{Self} \) (compare Chierchia’s (1988) syntactic reinterpretation of Bach and Partee 1980). We provisionally assume that this function is satisfied if it finds a suitable referring expression, but return to the issue in section 11. What is relevant here is that the c-command restriction on binding follows (see (19a–b)).\(^7\) NPIs form another class of dependent categories (Progovac 1994). The function they introduce, \( \text{Neg} \), looks for a polarity operator. As expected, the c-command restriction holds (see (19c–d)).

(19) a. I expect [John [to defend himself]].
   b. *I expect [himself [to defend John]].
   c. [No book [that John wanted to buy]] was on offer in any shop.
   d. *[The book [that no one wanted to buy]] was on offer in any shop.

Predicates introduce a thematic function \( \text{Th} \), satisfied by their subject.\(^8\) Indeed, predication requires c-command (Williams 1980), as shown by the Dutch depictives in (20). The depictive in (20a) can be associated with either subject or object, since both c-command it. By contrast, (20b) has only a subject-oriented reading, as the object does not c-command the depictive. Finally, (20c) is ruled out since the depictive is attached too high.\(^9\)

(20) a. dat [Jan\(i\) [Marie\(j\) [naakt\(i\), \(j\) ontmoette]]]
   that Jan Marie nude met
   ‘that Jan met Marie nude’
   b. dat [Jan\(i\) [naakt\(i\), \(j\) [Marie\(j\) ontmoette]]]
   c. dat [naakt\(i\), \(j\) [Jan\(i\) [Marie\(j\) ontmoette]]]

\( \theta \)-roles involved in complementation must be functions as well. Hence, in a double object construction two internal thematic functions are copied upward from the verb. The first is satisfied by

---

\(^7\) Some counterexamples are instances of logophoricity. Others, such as Dutch (i), involve binding by a silent argument, here a possessor. Passive, as in (ii), provides another environment in which an anaphor can apparently be related to a non-c-commanding category. Examples of this type can be understood if a hidden syntactic subject is assumed, which is interpretively related to the DP in the by-phrase (Baker, Johnson, and Roberts 1989). This proposal has the advantage of restricting apparent violations of c-command to constructions like passive, rather than allowing them anywhere.

(i) \[Die foto van zichzelf,] heeft geleid tot Jan’s ondergang.
   that picture of himself has led to Jan’s downfall

(ii) Er wordt [door Jan,] teveel over zichzelf, gepraat.
   there is by Jan too much about himself talked

\(^8\) If Chomsky’s (1986b) Visibility Condition is correct, thematic functions look for Case. Formally, \( \text{Th}(x) = x \), where the domain is \{nom, acc, dat, . . . \}.

\(^9\) This point is not invalidated by examples like (i) and (ii). Dutch has an operation of \( \hat{\mathrm{A}} \)-scrambling that is licensed by focusing of the moved element (Neeleman 1994). So, in (i) the depictive could be moved from a position c-commanded by \( \text{Jan} \). Examples in which secondary predication targets the absorbed subject of a passive can be understood along the lines of footnote 7.

(i) Ik denk dat NAAKT\(i\), Jan, Marie \(t\) niet heeft willen ontmoeten.
   I think that nude Jan Marie not has want meet
   ‘I think that Jan did not want to meet Marie nude.’
the verb’s sister; the second by the sister of a verbal projection. Thus, the Larsonian representation in (21) holds of English double object constructions.

(21) \[
\begin{array}{c}
\text{D} \\
V[Th] \\
V[Th, Th] \\
V[Th, Th] \\
D
\end{array}
\]

Internal θ-role assignment requires that the argument c-command the verb. This is confirmed by the observation that a verb cannot assign a θ-role to an argument contained in its complement (“exceptional θ-marking”). A verb need not c-command its complements, however, given that copying of thematic functions is recursive. This prediction is borne out by double object constructions, while copying also allows a base-generation analysis of scrambling (Neeleman 1994, Neeleman and Weerman 1999).

That dependencies as different as c-selection, movement, binding, NPI licensing, predication, and complementation adhere to the same requirement must be a consequence of the way the syntax operates, as proposed here.

### 3.3 Copying of Features versus Copying of Functions

One clarification is in order. The feature that satisfies a function can itself have been copied. The simplest such case is presented by c-selection, as the categorial features of a selected complement will in many cases have been copied upward under projection.

This observation does not threaten the explanation of c-command restrictions offered here. Copied features become properties of the node they are copied to. This fact restricts copying of features to nodes with compatible content. Categorial features, for instance, cannot be copied to a node that has a different set of categorial features.

Of course, copied functions also become properties of the node they are copied to. However, whereas features indicate what a node is, functions indicate what a node asks for. In many cases a node cannot be two things at the same time (say, a verb and a preposition). But it can be one thing (say, a verb) and ask for another (say, a prepositional complement). Thus, copying of functions is less restricted than copying of features.

### 3.4 C-Command in Other Frameworks

As explained previously, our proposals can be implemented in LFG, GPSG, and HPSG, which implies that in these theories, too, c-command restrictions can be derived (in LFG the relevant notion would be f-command).

CG is also based on function application, but does not assume a tree-based structural description of this operation. C-command restrictions on grammatical dependencies are therefore hard to capture. For example, it is possible in CG to combine the verb with its subject before it is
combined with its object. Hence, if c-command is calculated from the order in which constituents combine, an anaphor contained in the subject could be bound by the object.

Steedman (2000a) improves on this situation by mapping CG-style representations onto a constant predicate-argument structure: whether the object combines with the verb before or after the subject, the subject will be more prominent in the associated predicate-argument structure. Of course, c-command can be defined over predicate-argument structures (cf. Bach and Partee’s (1980) notion of F-command), and the crucial data can be captured by requiring it for binding, and so on. However, c-command requirements cannot be derived, precisely because in CG function application is conditioned linearly.

4 Obligatoriness

4.1 Deriving Obligatoriness

Koster’s (1987) configurational matrix includes obligatoriness. Functions, on his view, are undefined unless they acquire a value. Since the metarelation R, which establishes grammatical dependencies in Koster’s theory, is a function, obligatoriness follows.

Although the basic insight of Koster’s proposal is compelling, it is not obvious why unsatisfied functions should be undefined. Indeed, other frameworks based on function application simply stipulate that the root node may not contain unsatisfied functions (cf. LFG’s Functional Completeness). It is true, though, that unsatisfied functions cannot have syntactic or interpretive effects. Hence, some notion of economy suffices to make function application obligatory. We adopt (22) for this reason.

(22) Economy

Information in a node must be motivated in its immediate environment (as defined by Accessibility).

What counts as motivation for a particular bit of information depends on the effects the grammar allows it to have. For now, we adopt (23).

(23) Motivated information

a. Features: A feature in a node n is motivated if (i) it is interpreted in n, or (ii) it is copied.

b. Functions: A function in a node n is motivated if (i) it is satisfied in n, or (ii) it is copied.

Thus, a particular unit of information in a node is licensed by being copied or by having its distinctive effect in that node. This implies that once a function is introduced, it must give rise to a grammatical dependency: if a function is copied to the root node, its presence there is motivated only if it is satisfied, as upward copying from the root node is impossible. Dependent elements must therefore take an antecedent.

Indeed, the examples in (24) are all ungrammatical. The c-selecting verb in (24a) does not find a PP; the trace in (24b) does not find a moved element; the anaphor in (24c) does not find
a suitable antecedent; the NPI in (24d) is not in the c-command domain of a negative operator; and, finally, the predicate *naakt* in (24e) fails to take a subject.

(24)  
a. *John looked Mary.  
b. *John bought t.  
c. *I expect [you [to defend himself]].  
d. *Anyone better leave town.  
e. *dat [Jan, Marie] naakt ontmoette]  
    that Jan Marie nude met

As in the case of c-command, these data can be captured by stipulating for each of the grammatical dependencies involved that it is obligatory, but this does not explain the pervasive nature of this requirement.

4.2 Minimal Motivation

Economy, as generally understood, requires maximally simple representations. However, (22) freely allows a single bit of information to have multiple effects, even though this complicates the syntax, at least locally. We therefore strengthen (22) as follows:

(25)  
Economy (revised)  
Information in a node must be minimally motivated in its immediate environment (as defined by Accessibility).

The effects of (25) are twofold. First, a feature \( F \) in a node \( n \) is interpreted on \( n \) or copied to the dominating node, but not both. Hence, multiple interpretation of a single feature is impossible. For example, a percolated negative feature is only interpreted in its topmost incarnation. Second, a function \( \mathcal{F} \) in \( n \) must be input to copying or function application, but (normally) not both. Once satisfied, the function in \( n \) need not be copied, and vice versa. This precludes multiple satisfaction of a single function. For example, a \( \theta \)-role assigned to one argument cannot be reassigned to another.

5 Nodes

5.1 The Internal Structure of Nodes

Nodes are not atomic objects but collections of features and functions. This raises the question how information in a node is structured. Our basic assumption is as follows:

(26)  
Distinctness  
The syntax interprets attributes of a node that cannot be distinguished as one and the same.

Functions that look for different kinds of arguments and features that receive different interpretations are differentiated inherently, but attributes with identical properties require differentiation through the internal organization of the node.
Consider how a verb with two \( \theta \)-roles can be distinguished from a monadic verb. By Distinctness, (27a) and (27b) are identical nodes. (We place information contained in nodes between braces.)

(27) a. \( \{ Th, Th \} \)
b. \( \{ Th \} \)

One could argue that semantic labels like agent and theme are visible in syntax, so that \( \theta \)-roles are distinguished by type. But this hypothesis lacks empirical justification and has the conceptual drawback of introducing interpretive notions into the syntax.

Alternatively, one could analyze nodes on a par with Chomsky’s notion of numeration. In a numeration each lexical item carries an index indicating how many times it occurs. Thus, a dyadic predicate could have the following representation:

(28) \( \{ \ldots, Th_2, \ldots \} \)

This method of structuring nodes states the number of tokens of a particular attribute but provides no way of telling them apart. Hence, tracking the two thematic functions in (28) as they are copied upward is impossible. Similarly, one cannot determine which function is linked to which bit of semantic information in a terminal’s lexical entry.

These problems are circumvented if a node, in addition to its substantive properties, contains a tier that provides the basis for its organization. The fact that the two \( \theta \)-roles in (29) are linked to different positions in the ordering tier distinguishes them and makes them identifiable in copying and mapping to lexical semantics.

(29) \( \{ Th, Th \} \)

\[
\begin{array}{c|c}
1 & 2 \\
\end{array}
\]

We will argue in section 6.2 that Economy minimizes linking. Since two identical attributes can be told apart by linking only one, the number of links will usually be one fewer than the number of attributes to be distinguished. Thus, a transitive verb is represented as in (30a), while a triadic verb has the structure in (30b). In the interest of uniformity of presentation, we henceforth represent linking as minimal.

(30) a. \( \{ \ldots, Th, Th, \ldots \} \)

\[
\begin{array}{c|c}
1 & \phantom{1} \\
\end{array}
\]

b. \( \{ \ldots, Th, Th, Th, \ldots \} \)

\[
\begin{array}{c|c}
1 & 2 \\
\end{array}
\]

Distinctness, in conjunction with other considerations, thus leads to the conclusion that identical attributes in a node are distinguished by minimal linking to an ordering tier.

Ordering must originate in a terminal. Suppose that two identical functions are copied to a
node $\alpha$. These functions can coexist in $\alpha$ only if at least one of them is linked to $\alpha$’s ordering tier. But structuring $\alpha$ on an ad hoc basis would violate Inclusiveness: a property would be introduced in $\alpha$ that cannot be recovered from its daughters. So, when functions in a nonterminal node are ordered, this order must have been inherited from a daughter, which implies that it must ultimately originate in a head.

The above conclusions are familiar from GB $\theta$-theory, according to which predicative categories have a thematic grid that is introduced by a head.

5.2 Copying and Projection

The ordering tier was introduced as a basis for a node’s internal organization. This implies that order is a property of nodes, rather than of individual functions. An attribute can therefore carry along its position in the tier only if the tier is copied as well. Conversely, the attribute’s link will not be present in the dominating node if the attribute is copied individually. The tier seems an indivisible unit for copying.

It differs in this respect from the collection of substantive properties. Satisfied functions are stranded by copying of their unsatisfied counterparts. Similarly, external functions are at some point copied individually, stranding internal ones.

A node’s category is a formal property: despite some semantic correlations, there is no interpretive rule for categorial features. Given that such features also do not impose selectional requirements, they are most naturally analyzed as part of the ordering tier. The representations in (30) should therefore be modified as in (31). (We continue to use the simpler notation in (30) when categorial features are irrelevant.)

\[
(31) \quad \text{a. } \{ \ldots, \bar{F}, \bar{S}, \ldots \} \\
| \\
\langle +N, -V \ 1 \ \rangle \\
\text{b. } \{ \ldots, \bar{F}, \bar{S}, \bar{S}, \ldots \} \\
| \\
\langle +N, -V \ 1 \ 2 \ \rangle
\]

Copying targets all unsatisfied functions and all uninterpreted features in a node. If any of these attributes are linked and order is to be maintained, the tier must be copied as well. We refer to this as en bloc copying, in contrast to individual copying of attributes. Since categorial features are part of the tier, projection reduces to en bloc copying. A projection line ends if copying targets one or more individual functions; a projection line is extended whenever en bloc copying takes place (cf. Williams 1994 and Brody 1998a).

We are now in a position to sharpen an earlier conclusion. Copying of structure to a dominating node implies projection. In conjunction with Inclusiveness, this leads to the conclusion that order in a nonterminal node $\alpha$ is introduced by the head of $\alpha$.

Our theory of nodes will enable us to derive the third and fourth properties of the configurational matrix: uniqueness of the antecedent and nonuniqueness of dependents.
5.3 Nodes in Other Frameworks

Our framework treats nodes differently than others do. CG equates syntactic objects (‘‘nodes’) with (possibly complex) functions rather than with collections of features and simplex functions. CG can therefore not be transparently related to the proposals put forward here.

In unification-based theories, such as LFG, GPSG, and HPSG, similar attributes are distinguished in two ways. They may carry different labels (direct encoding), or they may be differentiated by means of embedding (hierarchical encoding). LFG represents θ-grids through direct encoding. A verb like show, for instance, contains the information in (32). GPSG also uses direct encoding, although implemented differently.

(32) \{\text{PRED SUBJ, OBJ, OBJ2}\}

Johnson (1988) discusses some drawbacks of direct encoding. In addition to these, direct encoding increases the expressive power of theories because every extra label in principle allows rules that refer to it. This places a heavy emphasis on the inventory of labels.

HPSG employs hierarchical encoding. The three arguments of show are part of a subcategorization list, which may be represented as in (33) (cf. Grimshaw 1990).

(33) \{\text{SUBCAT θ, }\{θ, \{θ\}\}\}

Hierarchical encoding and linking are similar in that θ-roles are not distinguished by name. The two methods are not identical, however. The subcategorization list in (33) groups the second and third θ-roles together. This group could be targeted by copying (or inheritance). No such operation is available in a theory based on linking, since there is no equivalent group. Thus, linking has less expressive power and is preferable on minimalist grounds. Corroborating empirical evidence is presented in sections 6 and 7.

6 The Uniqueness of the Antecedent

6.1 Deriving Uniqueness

Although external grammatical dependencies are unique, a head can entertain multiple internal dependencies. The latter possibility follows from the fact that a head can impose an order on its functions, which is maintained under projection. Let us now consider what happens if a function is satisfied externally to the projection in which it originates.

To be externalized, a function must be copied individually, as en bloc copying extends the projection. It follows that no maximal projection can enter into the same external dependency twice. Suppose that in (34) the two identical functions $\mathcal{F}$ in X, a maximal projection, are copied to $\delta$. These functions must be ordered in $\delta$ if they are to be distinguished. However, order is maintained only under en bloc copying, while the two functions are copied individually. Any order in $\delta$ must hence be introduced on an ad hoc basis, indicated here by a link to $1'$ rather than 1. Thus, (34) violates Inclusiveness. It follows that a dependent can enter into the same external dependency only once. Note that this explanation of the uniqueness of external dependencies is
available only if dependencies are established through upward copying and downward application, as required by Inclusiveness and Accessibility. If application were conditioned by sisterhood, one of the functions in X could be satisfied by $\beta$, while the other could be copied and applied to $\alpha$, with the result that X could enter into the same external relation twice.

$$\text{(34)} \quad \gamma \{\mathcal{T}_n\}$$

$$\begin{array}{c}
\alpha \\
\delta \{\mathcal{T}, \mathcal{T}_n\} \\
\beta \quad X \{\mathcal{T}, \mathcal{T}\} \\
\end{array}$$

As expected, the trace in (35a) cannot license two antecedents. Similarly, neither the reflexive in (35b) nor the depictive in (35c) can be associated with two arguments.

$$\text{(35) a. *What did John wonder what he bought?}$$

$$\text{b. *John confronted Mary with each other.}$$

$$\text{c. *(dat [Jan, [Marie [naakt [ontmoette]]]])}$$

As expected, the trace in (35a) cannot license two antecedents. Similarly, neither the reflexive in (35b) nor the depictive in (35c) can be associated with two arguments.

The uniqueness of the subject in predicational dependencies also holds for primary predicates. As remarked before, a verb may select multiple complements, but it cannot take more than one subject. The verb give, for example, contains three $\theta$-roles ordered through linking. The assignment of the internal $\theta$-roles requires that en bloc copying take place at least twice, after which the external $\theta$-role is copied in isolation. There is no triadic verb, however, that has two subjects and a single object.

### 6.2 The Lexical Encoding of Externality

We have assumed that linking is minimized as a result of Economy. However, (23), which determines the scope of Economy, does not mention linking. We therefore add (36), abstracting away from the option of linked features.

$$\text{(36) Motivated information (addendum to (23))}$$

**Linking:** A link to the ordering tier in a node $n$ is motivated if (i) the linked function is satisfied in $n$, or (ii) the link is copied.

As argued, any link that associates an external function with the ordering tier is lost under individual copying. According to (36), such loss of a link violates Economy, so that a linked role cannot be externalized. Conversely, an external function cannot be linked.
A simple way of marking functions as internal is implied—namely, by linking them. The syntax exploits this fact by interpreting unlinked functions as external. This follows from the Elsewhere Condition, which states that a specific rule takes precedence over a more general rule, where it can be applied. The specific rule would correspond to linking, while the general rule would correspond to absence of linking. For internal functions linking is available, so absence of linking, the general option, is blocked.

So linking is minimized, with unlinked roles interpreted as external. In (30a–b) the leftmost \( \theta \)-role must be external. Similarly, the single \( \theta \)-role of an unergative verb must be represented as in (37a), not (37b). Economy requires the \( \theta \)-role in (37b) to be internal. (37b) therefore represents an unaccusative verb.

\[
\begin{align*}
\text{(37)} & \quad \text{a. } \{ \ldots, \textit{Th}, \ldots \} \\
& \quad \text{b. } \{ \ldots, \textit{Th}, \ldots \} \\
& \quad \text{1}
\end{align*}
\]

In combination with Full Interpretation, the above proposal explains why c-selection is internal. The functions we have introduced all feed into the LF interface. For instance, if a \( \theta \)-role is assigned to a DP, the interface maps that DP to a semantic argument. C-selection is an exception. But c-selectional functions must be associated with a \( \theta \)-role: there is no c-selection of adjuncts. This suggests that, to satisfy Full Interpretation, such functions are associated with functions that do have an interface effect. As Kriszta Szendrői (personal communication) suggests, this association cannot be direct (functions are not linked to functions). Rather, the thematic and c-selectional functions must be linked to the same position in the tier.

\[
\begin{align*}
\text{(38)} & \quad \{ \ldots, \textit{Th}, \textit{Cat} \ldots \} \\
& \quad \text{1}
\end{align*}
\]

Since external \( \theta \)-roles are not linked, they cannot be associated with c-selectional functions, so that no predicate can determine the shape of its external argument.\(^{10}\)

In sum, a head can have multiple internal functions of the same type, but external functions are unique. Finally, c-selection of external arguments is ruled out.

### 7 The Nonuniqueness of Dependents

#### 7.1 Function Identification

Although external functions are unique, it does not follow that antecedents are associated with a single dependent. In fact, no limit is imposed on the number of dependents.

\(^{10}\) This conclusion carries over to subjects of passives and unaccusatives on the analysis of NP-raising presented in section 8.
Nodes can contain the same function twice only through linking. Now consider $\gamma$ in (39), whose daughters both contain the function $F$. Suppose $F$ is copied from both $\alpha$ and $\beta$. Ordering on $\gamma$ would be ad hoc, given that the copied information originates in different nodes. Since the two functions cannot be distinguished in $\gamma$, the resulting node is identical to one containing that function once. We call this function identification.

Function identification has an important consequence: given that the functions copied from $\alpha$ and $\beta$ collapse in $\gamma$, the antecedent that satisfies the function in $\gamma$ will in effect satisfy the functions in $\alpha$ and $\beta$. This is shown in (40).

We thus derive that the dependent in a grammatical dependency need not be unique. In (40) $\epsilon$ satisfies only one function in $\delta$, yet it is associated with two dependent categories. It follows that an antecedent can enter into the same grammatical dependency more than once.

### 7.2 Effects for Grammatical Dependencies

There is a wealth of data corroborating function identification. Although a reflexive never takes more than one antecedent, (41a) demonstrates that an antecedent can license more than one reflexive. Similarly, a negative operator can license more than one NPI (see (41b)).

(41) a. They introduced each other’s financial advisors to each other’s lawyers.
    b. John didn’t introduce anyone’s financial advisor to anyone’s lawyer.

In the same vein, an antecedent can bind two traces. Parasitic gaps, as in (42), are a case in point. There are many questions surrounding the parasitic gap phenomenon, but something like function identification must presumably be part of its analysis.

(42) These are the men that the police arrested $t$ without notifying $e$.

Further examples of function identification involve across-the-board phenomena. As is well known, a single antecedent can be associated with multiple dependents in coordinated structures.
This is true of movement, internal and external \( \theta \)-role assignment, NPIs, and binding. In our terms, each conjunct introduces a function, while the coordinate structure as a whole has only one.

### 7.3 A Reformulation of the \( \theta \)-Criterion

In coordinate structures a DP can satisfy more than one thematic function. However, the argument for function identification based on \( \theta \)-theory can be strengthened if we turn to secondary predication.

There is a basic bifurcation in the set of functions introduced so far. Some have a licensing capacity, others do not. An argument, for example, can be combined with a predicate only if it is assigned a \( \theta \)-role. Similarly, landing sites of movement must be dominated by a \( \text{Move} \) function. This contrasts with a function like \( \text{Neg} \): negative operators can be part of a tree in the absence of this function. The same is true of antecedents in binding dependencies. (See section 11.6 for further discussion.)

The distinction between licensing and nonlicensing functions allows us to formulate (43).

(43) *Exclusivity*

Consider a structure \([ \gamma \alpha \beta ]\). Let \( n_\alpha, n_\beta, \) and \( n_\gamma \) be the number of unsatisfied licensing functions in \( \alpha, \beta, \) and \( \gamma \), respectively. If a licensing function is satisfied in \( \gamma \), then \( n_\gamma \) may not be less than \( n_\alpha + n_\beta - 1 \).

(43) expresses that application of a licensing function blocks application as well as identification of other licensing functions. Exclusivity thus subsumes both the \( \theta \)-Criterion and the ban on movement to a \( \theta \)-position. Among the data captured by (43) is (44), ungrammatical under the reading that John showed himself to himself in the mirror.

(44) *John showed himself (in the mirror).

(44) has two representations. In the first, the object satisfies two \( \theta \)-roles in its mother (the goal and theme). This one-step reduction of unsatisfied licensing functions by two gives rise to (45), a representation usually ruled out by the \( \theta \)-Criterion.

(45) *

\[
\begin{array}{c}
V \{ Th, Th#, Th# \} \\
V \{ Th, Th, Th \} \quad D \\
1 \\
2
\end{array}
\]

Alternatively, (44) could be a case of raising to \( \theta \)-position. The function encoding the verb’s theme would be assigned to a trace bound by the goal DP. This results in (46), where the goal DP satisfies two licensing functions. This one-step reduction of unsatisfied licensing functions by two again violates (43).
Let us now turn to cases in which an argument is associated with both a primary and a secondary predicate. In (47), for example, *Marie* is assigned the verb’s internal θ-role, as well as the external θ-role of *naakt* ‘nude’.

(47) dat [Jan [Marie] [naakt ontmoette]]

that Jan Marie nude met

Most proposals in the literature complicate syntactic theory to accommodate such data. Essentially, they stipulate that the θ-Criterion holds of primary predication only. Williams (1983) and Chomsky (1986b) do so by amending the θ-Criterion itself. Stowell (1981, 1983) proposes that secondary predication is mediated by a PRO subject, which necessitates otherwise redundant assumptions about control (Williams 1994).11

These complications dissolve as a result of function identification. The AP’s external thematic function in (47) is copied into a node that also contains the verb’s θ-grid. The θ-role copied from AP cannot be ordered with respect to those in the verb’s θ-grid. Hence, it must collapse with one of the verb’s θ-roles, the internal one in (47), as shown in (48).

(48) V {... *Tθ*}

D V {..., *Tθ*} A {*Tθ*} V {..., *Tθ*}

The adjective’s θ-role can also be identified with the verb’s external θ-role, giving rise to a subject-oriented variant of (47).

So, if secondary predication is an instance of function identification, the problems with the formulation of the θ-Criterion referred to above do not arise. (48) adheres to Exclusivity, since D satisfies a single licensing function in its mother.

---

11 “Vertical binding” allows unification (in the sense of Williams 1994) of VP’s external θ-role with the θ-role of a subject-oriented secondary predicate. Hence, the θ-Criterion need not be revised for such structures. The account does not carry over to object-oriented secondary predication, as it can only involve external properties of the vertical binder.
7.4 Prenominal Modification

Additional evidence for function identification comes from (intersective) prenominal modification. Consider (49).

(49) the green door

Green is a predicate. But how its external θ-role is satisfied is unclear, as there is no subject to which it can be assigned. Since predication cannot target a dominating node, the DP that contains green does not qualify. One could assume a PRO subject for the AP, as in the [PRO green] door (Van Gestel 1986). However, PRO cannot be dependent on a dominating node either, so that this proposal recreates the original problem.

Higginbotham (1985) suggests a way of interpreting examples like (49). He proposes that NPs are predicates whose θ-role is bound by a determiner if the DP is used as an argument. The θ-role of a prenominal modifier and the so-called R-role of the noun are related by coindexation. So, whatever satisfies the R-role also satisfies the θ-role of the modifier. The green door therefore refers to something that is both green and a door.

Function identification captures the effects of Higginbotham’s analysis without coindexation. In (50) both N and A contain a thematic function that is copied onto the dominating NP. The adjective’s role cannot be ordered with respect to the R-role of the noun and consequently they collapse. Whatever satisfies the thematic function in NP will satisfy the thematic functions of both the noun and the adjective, as required.

(50) N {Th}  
    A {Th}  
    N {Th}

So, our theory provides a unified account of prenominal modifiers and secondary predication. Many alternative theories need two processes by which θ-roles can be satisfied: assignment for predication and coindexation for prenominal modifiers.

7.5 Uniqueness in Other Frameworks

Section 5 identified two alternatives to linking: labeling and embedding. θ-roles, for example, are labeled in LFG and GPSG, but distinguished through embedding in HPSG. Can theories using these alternatives explain why XP dependents take a unique antecedent, while antecedents can be related to multiple dependents?

We first consider the uniqueness of external dependencies. If labels distinguish grammatical functions, this fact cannot be captured, as after externalization such functions are still different. In Kaplan and Bresnan 1982, for instance, SUBJ is realized VP-externally, while OBJ and OBJ2 are projected within the VP. However, the absence of an external SUBJ2 function is coincidental.

HPSG distinguishes identical functions by embedding. In (51) α is a unit in NODE and can consequently be inherited by the structure dominating NODE, even if this structure is not projected.
from it. One can think of $\alpha$ as complex—on a par with \textsc{subcat} features—but not phrase-bound—on a par with slash features.

\[(51) \{\text{node} \ldots \{_{\alpha} \mathcal{F}, \{_{\alpha} \mathcal{F}\}\ldots\}\} \]

But if embedding allows order to be maintained externally to a head’s maximal projection, the uniqueness of external dependencies cannot be derived.

As regards the nonuniqueness of dependents, theories based on unification look promising. Identical features inherited from two daughters unify in the mother node. Hence, the analyses developed here should be available in such theories. Indeed, our analysis of parasitic gaps is equivalent to the GPSG/HPSG account.

That $\theta$-roles are labeled in LFG makes a unification-based analysis of secondary predication impossible, however. Labeling combined with unification incorrectly predicts that multiple dependents must carry the same function. A \textsc{subj} role could be unified with a \textsc{subj} role, but not an \textsc{obj} role. LFG consequently resorts to equality statements to capture object-oriented secondary predication, thereby failing to exploit the advantages of unification.

The same holds of HPSG. Encoding $\theta$-roles in a single \textsc{subcat} feature blocks identification of individual roles. The monadic \textsc{subcat} feature of a depictive could be unified with the \textsc{subcat} feature of a verb if the latter is monadic as well, but a simple unification-based account of object-oriented depictives is not possible. As in LFG, extra statements are needed to unify part of one \textsc{subcat} feature with part of another.

The above theories recognize external and internal dependencies. GPSG and HPSG, for example, distinguish head and foot features. No such distinction is made in CG as it lacks phrase structure. Consequently, the uniqueness of external dependencies cannot be captured.

Since CG is not based on unification, associating an antecedent with multiple dependents requires specific combinatory statements. Steedman (2000b) suggests that these all fit the same schema, but even so, multiple dependents can be accommodated only by rule.

7.6 Summary and Preview

Economy, Inclusiveness, Accessibility, and Distinctness explain four properties of the configurational matrix (c-command, obligatoriness, uniqueness of antecedents, and nonuniqueness of dependents). This leaves the final property, locality.

A discussion of locality should include movement, but this requires a closer look at the nature of traces. Section 8 therefore deals with NP-trace, while $\tilde{A}$-trace and $X^0$-trace are discussed in sections 9 and 10. We turn to locality in section 11.

8 NP-Trace

8.1 A Uniform Analysis of Derived and Nonderived Subjects

So far we have implicitly assumed two notions of subject. Transitive and unergative subjects are licensed by thematic functions, while passive and unaccusative subjects are licensed by $\text{move}$.
One can generalize over these cases by assuming a structural subject position, but since the notion of external argument was independently motivated above, this would be theoretically anomalous.

The problem originates in the premise that traces uniformly introduce \textit{Move}. Three sets of data suggest, however, that this is not true of NP-trace. As Burton and Grimshaw (1992) show, it is possible to coordinate predicates with VPs containing the trace of A-movement. Dutch, for instance, allows coordination of an AP predicate and a passive VP.

\begin{equation}
\text{(52) Jan vertrok \{[AP dronken] en [VP door iedereen t verraden]\}.}
\end{equation}

Jan left drunk and by everybody betrayed

The Coordinate Structure Constraint states that coordinates must be of the same semantic type (but not the same syntactic category; Sag et al. 1985). So, given that the AP in (52) is a predicate, the passive VP must be a predicate as well. In our terms the AP and the VP must introduce the same function, namely, an external \(\theta\)-role. After coordination these functions undergo identification and the resulting role is assigned to \textit{Jan}.

If the VP in (52) introduces a \(\theta\)-role, we must account for this. As the passive verb’s single thematic function is satisfied by the trace, the relevant \(\theta\)-role cannot have been copied from \textit{V}. It must therefore find its source in the trace. This echoes Williams’s (1986, 1994) claim that NP-trace is a syntactic means of externalizing an internal \(\theta\)-role.

Corroborating evidence comes from prenominal modification. Prenominal modifiers are interpreted through function identification. Their external \(\theta\)-role is identified with the noun’s R-role. This implies that passive participles derived from unergatives cannot be used prenominally. If passive absorbs the verb’s external (and only) \(\theta\)-role, no thematic function remains that can be identified with that of the noun. Indeed, Dutch freely allows impersonal passives, but impersonal passive participles are barred from occurring prenominally (Perlmutter 1978, Hoekstra 1984, Ackema 1999). By contrast, passive participles of transitive verbs can occur in this position.

\begin{equation}
\begin{aligned}
(53) & \text{a. Er wordt (door Jan) overal geslapen.} \\
& \text{there is (by Jan) everywhere slept} \\
& \text{‘Jan sleeps anywhere.’} \\
& \text{b. *de (door Jan) geslapen hond} \\
& \text{the (by Jan) slept dog} \\
& \text{c. de (door Jan) t geslagen hond} \\
& \text{the (by Jan) beaten dog}
\end{aligned}
\end{equation}

Apparently, the prenominal modifier in (53c) does have an unsatisfied \(\theta\)-role. But since the single \(\theta\)-role of a passive participle is assigned internally, it must again be NP-trace that contributes an external thematic function, here identified with that of the noun.

Our third argument is based on secondary predication. The gerund depictive in (54a) can be associated with the subject but not the object of a transitive verb (it is attached too high). Yet, gerunds may be related to passive and unaccusative subjects, as (54b) shows. But if NP-trace introduced \textit{Move}, Exclusivity would rule out these examples, as the raised DP would satisfy both the gerund’s \(\theta\)-role and a \textit{Move} function.
(54) a. Johni killed Billj [thinking about Mary].
   b. Billi was killed/died t [thinking about Mary].

If NP-trace contributes a thematic function, however, this function will be identified with the gerund’s external θ-role, so that the raised subject satisfies exactly one licensing function in the node that dominates it, as required.

That NP-trace should introduce a θ-role is not surprising: thematic functions and the function introduced by the trace of A-movement look for the same kind of object (a syntactic argument). The null hypothesis, then, is that these functions are identical.

8.2 The Anatomy of Raising

Let us consider NP-raising in more detail. (55) is the structure projected by an unaccusative verb. The verb’s θ-role is copied to VP, where it is satisfied by XP. This phrase is either NP-trace (as in John arrives t) or a constituent containing NP-trace (as in John seems [t to have left]). By hypothesis, XP itself introduces a θ-role, which is copied to VP as well. This θ-role must be external in VP: it cannot collapse with any internal θ-role (for reasons explained below), and Inclusiveness forbids ad hoc linking. Thus, an unaccusative verb merged with a complement of the right type can still project a predicative VP.

(55) *
   \[
   \alpha \{ \text{Th}_\# \}
   \]
   \[
   D \quad V \{ \text{Th}, \text{Th}_\# \}
   \]
   \[
   V \{ \text{Th} \} \quad X \{ \text{Th} \}
   \]

Although NP-trace and secondary predicates both contribute a θ-role, they differ fundamentally with respect to Exclusivity. This principle rules out a node satisfying two licensing functions. However, it also restricts identification: if two licensing functions collapse, the number of unsatisfied licensing functions is reduced by one.

Crucially, NP-trace satisfies a θ-role in the dominating node. The implication is that its own θ-role cannot be identified with a θ-role in its mother, as that would lead to a second reduction in unsatisfied licensing functions. This is illustrated in (56), where XP either is or contains NP-trace. Thus, our earlier explanation for the ban on raising to θ-positions still holds, even if some Move functions are reanalyzed as θ-roles.

(56) *
   \[
   \quad \alpha \{ \text{Th}_\# \}
   \]
   \[
   D \quad V \{ \text{Th}, \text{Th}_\# \}
   \]
   \[
   V \{ \text{Th} \} \quad X \{ \text{Th} \}
   \]
A secondary predicate is not assigned a \( \theta \)-role, so the function it contributes is free to collapse with a function of the verb. This makes our account of the ban on raising to \( \theta \)-positions superior to the standard chain-based alternative (Chomsky 1986b, Brody 1998b), which cannot accommodate raising to a secondary \( \theta \)-position (see (54b)).

Our analysis also rules out raising to object position. This operation would require that NP-trace’s \( \theta \)-role end up as internal after one or more steps of copying. But this implies that it is either linked to the ordering tier or identified with a linked \( \theta \)-role. The former option is excluded by Inclusiveness, whereas the latter violates Exclusivity.

In sum, the proposal that NP-trace introduces a thematic function allows a unified characterization of subjects. Base-generated and derived subjects both satisfy an external \( \theta \)-role. The only difference is that the role of the former originates in a verb; that of the latter in a trace.\(^{12}\)

9 Ā-Trace

9.1 Toward an Inventory of Traces

The above discussion suggests that NP-trace is a very simple lexical item, containing a \( \theta \)-role plus the minimal specification required to function as an argument. The same cannot be true of Ā-traces: these do not introduce a \( \theta \)-role, as their antecedents do not occupy A-positions. Moreover, Ā-traces are not contained in the lexicon. Since an Ā-trace has all the syntactic properties of its antecedent, a different trace would have to be listed for every phrase that can be moved, not a tenable conclusion.

Indeed, in the copy theory of movement, traces are not lexical items. Movement is analyzed as a dependency between the moved element and a full copy of it. This dependency should be mediated by a function, but none of the lexical items that make up the lower copy can count as its source. This can be remedied by assuming a rule like (57), which corresponds to Move \( \alpha \) in GB Theory and Form Chain in minimalism.

\[(57) \text{Move Introduction} \]
\[\text{Add } \text{Move to a node.}\]

Although a \text{Move} Introduction rule seems unavoidable, combining it with the copy theory of movement goes against two core assumptions of our theory.

First, if traces have internal structure, the introduction of \text{Move} in the top node of a trace violates Inclusiveness. Second, if \text{Move} looks for a full copy of the trace, its satisfaction violates Accessibility whenever the moved element has internal structure. Crucially, \text{a picture of John} and \text{a picture of Mary} should not be treated as copies, but this can only be established by inspection of the internal structure of these phrases.\(^{13}\)

\(^{12}\) This view of subjects raises questions about the status of expletives. Weather \text{it} is not a true expletive but a pseudoargument (Chomsky 1981). The same holds of \text{it} in examples like \text{It is generally believed that Greeks love food} (Bennis 1986). The analysis of expletives associated with DPs is complex, but there are arguments for not analyzing them as occupying the subject position, at least not uniformly (Koeneman 2000).

\(^{13}\) Nunes (2001) proposes that identity of complex chain members is checked by inspection of the indices on the terminals they contain (cf. Chomsky 1995b, 2000). Such global comparison violates Accessibility.
We conclude that traces are terminals. The analysis we develop below is based on this view, in conjunction with a refinement of (57).

9.2 Terminals and Inclusiveness

GB-style lexical insertion implies that syntactic nodes contain not only syntactic, but also phonological and semantic information, which is stripped away at particular stages of the derivation. The alternative (Jackendoff 1997, Halle and Marantz 1993) claims that nodes contain only syntactic information, linked to phonological and semantic information through mapping rules. Minimalist assumptions favor the latter “separationist” view.

A lexical entry, then, is a mapping rule that associates minimal syntactic, semantic, and phonological representations. That is, a terminal is related at the interfaces to phonological and semantic matrices with the same “lexical address.” The terminal itself must be matched with its lexical entry through the same address (but see Samek-Lodovici 1999).

A¯-traces differ from other terminals in not having a lexical address (they are not lexical items). Therefore, to satisfy Inclusiveness, they must be related to a node from which their properties can be recovered. Move provides a way of doing so: it associates the trace with an address, albeit a syntactic rather than a lexical one. In (58) properties p₁, p₂, and so on, are licensed on t as δ is t’s syntactic address after function application and δ has these properties. The properties of t are not copied from δ; they are generated independently and licensed through the satisfaction of Move. In other words, this function guarantees identity of nodes in a way that parallels the matching of terminals with lexical entries.

(58)
\[
\gamma \{ \text{Move}_# \}
\]
\[
\delta \{ p_1, p_2, \ldots \} \quad \beta \{ \text{Move} \}
\]
\[
\alpha \quad t \{ \text{Move}, p_1, p_2, \ldots \}
\]

The presence of Move in the trace is reconciled with Inclusiveness if the rule that introduces it is sensitive to the absence of a lexical address. Let us therefore replace (57) by (59), which is comparable to the slash termination rules of GPSG.

(59) Move Introduction (revised)

[Address: -] ⇒ \{ ..., Move, ... \}

If address-less terminals trigger Move Introduction, the two problems identified earlier are eliminated: properties of traces are predictable, in keeping with Inclusiveness, and since Move does not look for a copy of a subtree, it need not access inaccessible structure.

\[14\] We abstract away from morphologically derived terminals, which, if not stored, obtain their properties through morphological principles.
9.3 Reconstruction and the Generalized Projection Principle

Two further advantages of our analysis of traces concern reconstruction and Brody’s (1995, 1998b) Generalized Projection Principle (GPP).

Reconstruction is an effect of syntactic address assignment. Move guarantees that a trace and its antecedent have identical properties. Consider NP-raising from this perspective. A-movement leaves behind a trace that introduces a thematic function, as shown by the data discussed in section 8. Such functions do not provide the dependent with an address. Hence, A-movement should differ from A¯-movement in not exhibiting (syntactic) reconstruction effects. Such effects are indeed absent.

The Dutch example in (60a) demonstrates that the trace of a topicalized anaphor behaves like an anaphor: it shares a Self function with the fronted element. Reconstruction is absent in (60b), however: the trace left behind by passivization does not share the Self function of its antecedent, as the ungrammaticality of (60b) indicates.

(60) a. Alleen zichzelf vindt Piet t aardig.
    only himself finds Piet nice
    ‘It is only himself that Piet likes.’

b. *Ik zie [zichzelf, Piet t getoond worden].
    I see himself Piet shown be
    ‘I see Piet being shown to himself.’

A second advantage of our analysis of traces relates to the GPP. This principle states that projectional (categorial, thematic, and selectional) requirements that link a chain to its environment must hold in and be satisfied by the chain root. Thus, there can be no movement to a θ-position and no selection (in a broad sense) from a derived position. The first restriction follows from Exclusivity (see sections 7.3 and 8.2).

The second restriction can be derived from the assumption that the antecedent is the trace’s address. This of course implies that the properties of a trace are recovered from its antecedent, but also that the properties of the antecedent must have their effect through the trace. The reason for this lies in Economy, which requires information in a node to be minimally motivated. More in particular, the notion of syntactic address yields a third way of meeting (25). Since attributes can have an effect by being the argument of an address-assigning function, we assume that this is sufficient to license them.

(61) Motivated information (revised)

a. Features: A feature in a node n is motivated if (i) it is interpreted in n, (ii) it is copied, or (iii) it is part of a syntactic address (n satisfies Move).

b. Functions: A function in a node n is motivated if (i) it is satisfied in n, (ii) it is copied, or (iii) it is part of a syntactic address (n satisfies Move).

c. Linking: A link to the ordering tier in a node n is motivated if (i) the linked function is satisfied in n, (ii) the link is copied, or (iii) the link is part of a syntactic address (n satisfies Move).
So, a function $F$ in a node $n$ satisfies Economy if the node dominating $n$ contains an address-assigning function by which the properties of $n$ are “transferred” to a trace. $F$ can therefore not be copied from $n$ or satisfied by a daughter of $n$. Selection from a derived position is ruled out as a result.

Binding demonstrates the ban on selection from a derived position. Dutch focus scrambling is the $\bar{A}$-movement that relates (62a) to (62b) (Neeleman 1994). As the data show, scrambling the constituent introducing $\text{Self}$ does not extend binding possibilities, contrary to what one would expect if copying were possible from a derived position.15

(62) a. dat Kimi alleen Mariej toestaat om zichzelfj/*i te fotograferen
   that Kim only Marie allows COMP herself to photograph
   ‘that Kim allows only Marie to photograph herself’

b. dat Kim, zichzelfj/*i alleen Mariej toestaat om t te fotograferen
   that Kim herself only Marie allows COMP to photograph
   (same)

Thematic functions do not give rise to address assignment. The consequences are twofold. First, as explained above, reconstruction is absent in NP-raising. Second, copying of functions should be possible from derived subjects. Independent conditions often prevent this, but in (63) a $\text{Self}$ function has indeed been copied from a raised NP. These two observations support the view that NP-trace differs fundamentally from $\bar{A}$-trace.

(63) Jani liet [zichzelf, Marie t getoond worden].
   Jan let himself Marie shown be
   ‘Jan let Marie be shown to him.’

10 X⁰-Trace

We consider head movement next. Such movement patterns with $\bar{A}$-movement as far as reconstruction is concerned. The trace of head movement has all the syntactic properties of the moved head. Thus, an X⁰-trace must introduce an address-seeking function.

(64) $Hd$-Move Introduction
    [Address: −] ⇒ \{ . . . , $Hd$-Move, . . . \}

Since head movement involves address assignment, our explanation of the GPP extends to heads in derived positions. Indeed, such heads cannot assign $\theta$-roles, and so on.

Head movement raises two questions. First, in the absence of bar levels, how can we distinguish head and XP movement? Second, on the first-branching-node definition of c-command implied by our theory, how can we allow head-to-head adjunction (Baker 1988, Bobaljik and Brown 1997)? Confronted with these questions, one might abandon head movement altogether.

15 It is sometimes argued that $\bar{A}$-movement does feed binding, but examples always involve picture nouns. There is independent evidence that anaphors contained in picture nouns behave like logophors (Reinhart and Reuland 1993).
(Cormack and Smith 1997, Brody 2000, Koopman and Szabolcsi 2000). However, we believe that the problems identified above are surmountable.

There is an obvious way in which heads can be distinguished from other nodes: the head of a node $\alpha$ is that daughter from which $\alpha$ is projected. Hence, the function that encodes head movement could look for a copy of the node that hosts it (modulo individually copied functions). Thus, in (65) $Hd-Move$ is satisfied by $\alpha$.

\[
\begin{align*}
(65) \quad & \alpha \{Hd-Move\} \\
\quad \alpha & \quad \beta \{Hd-Move\}
\end{align*}
\]

A function cannot be satisfied by the node in which it originates. Consequently, $Hd-Move$ cannot be satisfied by the lower instantiation of $\alpha$ in (66). It follows that the function can be copied up in the projection of the head that introduces it, as required.

The problem raised by head-to-head adjunction is more complex. Although $Hd-Move$ in (65) is satisfied by the lower copy of $\alpha$, this is not the node that should be linked to the $X^0$-trace, at least not in the case of head-to-head adjunction. A more complete structure is given in (67), where $\gamma$ must function as $\beta$'s address.

\[
\begin{align*}
(67) \quad & \alpha \{Hd-Move\} \\
\quad & \quad \beta \{Hd-Move, p_1, p_2, \ldots\} \\
\quad \gamma \{p_1, p_2, \ldots\} & \quad \alpha
\end{align*}
\]

How can the function satisfied by $\alpha$ in (67) have the effect that properties of $\gamma$ are reconstructed in $\beta$? The answer is dictated by Accessibility. It must be that the properties of $\gamma$ are present in its mother. We can arrange for this by allowing en bloc copying to embed $\gamma$ in $\alpha$. As a result, $\gamma$ is made accessible as an address for $\beta$ (see (68)).

\[
\begin{align*}
(68) \quad & \alpha \{\ldots, Hd-Move\}, \ldots\} \\
\quad & \quad \alpha \{\ldots, \{\gamma p_1, p_2, \ldots\}, \ldots\} \quad \beta \{Hd-Move, p_1, p_2, \ldots\} \\
\quad \gamma \{p_1, p_2, \ldots\} & \quad \alpha \{\ldots\}
\end{align*}
\]
At first sight either $\alpha$ or $\gamma$ could act as the address for $\beta$. However, information that forms an address for a trace cannot project and vice versa (see (61)). Therefore, the address for the trace must be the embedded node, $\gamma$.

We see embedding as a storage mechanism: $\gamma$ is stored in $\alpha$’s first projection. Some such mechanism is independently required. Without it, grammars would be context free, but Chomsky (1957), Huybregts (1985), and others have established that natural language exhibits some context-sensitivity. Storage, including its relation to cross-serial dependencies, is discussed in more detail in Neeleman and Van de Koot 2001.

11 Locality

11.1 Preliminaries

The stage is now set to address the fifth property of the configurational matrix, locality. Certain aspects of locality are not syntactic but originate in the interaction between syntax and syntax-external systems. For example, adjacency requirements can be captured by invoking PF mapping procedures (Emonds 1985, Neeleman and Weerman 1999). Performance systems may also give rise to locality effects. The contrast in locality between rightward and leftward movement, for example, does not follow from the (invariant) configurational matrix. Instead, two commonly assumed properties of the human parser, informational monotonicity and a filler-driven strategy for the creation of traces, can be held responsible (Rochemont 1992, Ackema and Neeleman 2002).

Other aspects of locality follow from our proposals, as we will now show.

11.2 Effects of Function Identification

Recall that if a function is copied to a node from both its daughters, identification takes place. This has implications for the domain in which functions can be satisfied. Consider a node $n_1$ that introduces a function $F$ satisfied in $n_2$. As a result of function identification, $n_2$ is a barrier for any other instance of $F$ copied into the path from $n_1$ to $n_2$. In other words, the maximal domain in which a function $F$ can be satisfied coincides with the first dominating node in which another instance of $F$ is satisfied.

A first illustration of this effect of function identification can be found in the domain of predication. External $\theta$-roles need not be assigned within the m-command domain of the predicate, as shown by (69), from Williams 1994. The external $\theta$-role of the AP *still explosive* is copied out of the projection headed by *while* into the VP, where it collapses with the $\theta$-role ultimately assigned to *the device*.

(69) The device arrived [while [still explosive]].

In view of this, it is surprising that such long-distance predication is unavailable in (70). Whatever controls PRO in these Dutch examples is also the semantic subject of *naakt* ‘nude’. Apparently, the external $\theta$-role of this predicate can only be assigned to PRO.
These data follow from identification. When the θ-role of a depictive is copied to a node that already has a θ-role, these roles collapse. Hence, the argument assigned the latter role must also be the subject of the depictive. In (70) the θ-role of naakt must thus collapse with the verbal θ-role assigned to PRO, as desired. But in (69) no identification takes place within the adjunct, because while does not have a θ-grid. Consequently, the depictive’s external role will not collapse with another thematic function until it reaches the VP headed by arrive. In general, then, predication can never cross a subject.

Aspects of Relativized Minimality (Rizzi 1990) can be derived from identification as well. Raising to subject is not possible across a subject. This fact receives the same explanation as the locality of predication. NP-trace introduces a thematic function. This function cannot be copied beyond a subject, as it would be identified with the θ-role assigned to that subject. The result is (71), where XP either is or contains an NP-trace.

> (71) *$V \{Th, Th_#\}$
> 
> $V \{Th, Th\}$
> 
> $X \{Th\}$

Even if (71) were grammatical, it would constitute a case of raising to a θ-position, rather than a case of superraising. But as we have shown, raising to a θ-position is ruled out by Exclusivity: the formation of VP in (71) involves an illegitimate reduction of unsatisfied licensing functions by two.

Function identification is also helpful in ruling out wh-movement across a fronted wh-operator. The function introduced by a wh-trace will be identified with another Move function, copied into its path. (73) is the representation of $t$ read $t$ in (72).

> (72) *What do you wonder [who $t$ read $t$]?

---

16 The secondary predicate must be attached lower than PRO (see section 11.6).

17 The parallel between the locality of secondary predication and NP-raising is also exploited by Williams (1994), who bases his analysis on “vertical binding.”
Even if (73) were grammatical, it would not give rise to wh-movement across a wh-operator, given that the two \textit{Move} functions collapse. But (73) in fact violates Exclusivity: the formation of $\alpha$ involves an illegitimate reduction of licensing functions by two. The subject satisfies a \(\theta\)-role of the verb, while its \textit{Move} function undergoes identification.

11.3 Effects of Function Application

A function is satisfied if it occurs in a configuration in which the daughter of its host is of the desired type. Since function application will take place whenever its structural description is met, a function traveling up the tree is satisfied at the earliest opportunity.

The fact that function application is automatic allows us to derive a further set of Relativized Minimality effects. Facts like (74) do not follow from identification, as only one \textit{Move} function is present. However, if \textit{Move} looks for an operator, and if \textit{whether} is classified as such, the ungrammaticality of (74) is explained.

(74) *What do you wonder [whether John read \textit{t}]?

Intervention effects depend on the typology of functions. We have assumed that \textit{Move} is an operator-seeking function, but it probably represents a class of functions distinguished by their arguments (Rizzi 2000a,b). The same goes for other dependencies. Local and long-distance anaphors may well introduce different functions. The latter are explored below; the former could themselves divide up into different classes.

Function application underlies three further locality conditions. The first is the Head Movement Constraint (HMC). \textit{Hd-Move} is satisfied in (75) but not in (76), since a function cannot be satisfied by the node that introduces it.

(75) \[ \alpha \{Hd-Move\} \]

(76) \[ \alpha \{Hd-Move\} \beta \]
The HMC follows from this. In (77) $\text{Hd-Move}$ is copied to $\beta$ and automatically satisfied there. It is thus impossible to relate $\alpha$ and $\gamma$ using this function.

\[
(77) \quad \begin{array}{c}
\alpha \\
\alpha \\
\beta \{\text{Hd-Move}\} \\
\beta \\
\gamma \{\text{Hd-Move}\}
\end{array}
\]

The automatic nature of application also explains why excorporation from a complex head is impossible for the adjoined category, but allowed for its host (Ackema 1999). Suppose that (75) and (76) represent a complex head rather than a head-complement structure (the order should then be $\beta$-$\alpha$). As indicated in (75), a $\text{Hd-Move}$ function introduced by $\beta$, the nonhead, will be satisfied by $\alpha$. Consequently, the nonhead cannot excorporate. By contrast, a $\text{Hd-Move}$ function introduced by $\alpha$, as in (76), will not be satisfied within the complex head, so that excorporation is possible.

Finally, the automatic nature of application explains the locality of anaphoric binding. At first sight it has no bearing on this dependency. The anaphor in (78a), for example, can be bound by $\textit{Bill}$ but also by $\textit{John}$. Apparently, the function $\textit{Self}$ can skip a potential antecedent. However, $\textit{Self}$ cannot skip just any potential antecedent: it cannot cross a subject (see (78b)).

\[
(78) \quad \begin{array}{a}
\text{a. John$_i$ showed Bill$_j$ himself$_{ij}$ (in the mirror).} \\
\text{b. John$_i$ expected Bill$_j$ to invite himself$_{ij}$/*$_i$.}
\end{array}
\]

Several authors have suggested that binding theory is parasitic on $\theta$-theory. Williams (1980) argues that binding is subject to the Predicate Opacity Condition, according to which an anaphor cannot occur free in a predicate. Reinhart and Reuland (1993) propose a theory of reflexivity in which coargumenthood plays a central role. Both proposals explain the contrast between (78a) and (78b).

In our theory these data can be understood if $\textit{Self}$ looks for an unsatisfied thematic function. This proposal goes back to Williams’s (1994) claim that anaphors take $\theta$-roles as antecedents, for which the Dutch example in (79) provides some corroboration.18

\[
(79) \quad \begin{array}{a}
\text{een [[zichzelf respecterend] mens]} \\
\text{a. himself respecting person}
\end{array}
\]

As argued earlier, the $\theta$-role of prenominal modifiers is identified with the noun’s R-role. (79)

---

18 Theories based on the idea that the antecedent of an anaphoric dependency is a $\theta$-role predict correctly that binding involves A-positions only.
hence does not contain a subject that can bind the anaphor, making it problematic for standard approaches to binding. But if anaphors take θ-roles for antecedents, zichzelf in (79) can rely for its interpretation on the external role of the prenominal modifier.

Let us now return to (78a). This example has the structure in (80).

\[
(80) \quad \alpha \{ \text{John}_# \} \\
D \quad V \{ \text{John}, \text{John}_# \} \\
\quad \quad \downarrow 1 \\
D \quad V \{ \text{John}, \text{John}_#, \text{Self}_# \} \\
\quad \quad \quad \downarrow 1 \ 2 \\
V \{ \text{John}, \text{John}, \text{Self} \} \quad D \{ \text{Self} \} \\
\quad \quad \quad \quad \downarrow 1 \ 2
\]

The function \textit{Self} is introduced by the direct object and satisfied in its mother by one of the θ-roles of the verb. There is some choice in this respect: it can be the external role (the apparent antecedent is the subject) or the role linked to position 1 (the apparent antecedent is the indirect object). The function linked to 2 is unavailable. Through \textit{Self} an anaphor’s interpretation is linked to a particular θ-role. However, if that role is assigned to the anaphor, an unacceptable circularity of interpretation results (Chomsky 1981).

(78b) under a construal of \textit{himself} as \textit{John} is excluded by the fact that \textit{Self} is satisfied automatically by the θ-role ultimately assigned to \textit{Bill}, once it is copied into the projection of \textit{invite}. Hence, it cannot be related to the θ-role that \textit{expect} assigns to \textit{John}.

This account of Principle A predicts that anaphor and apparent antecedent need not always be coarguments. This is the case if \textit{Self} is copied to a node whose daughter does not contain a θ-role that could satisfy it. Suppose that \textit{Self} is copied from an exceptionally Case-marked subject. The only unsatisfied θ-role in the sister of such a subject is the one assigned to it, but this role is not a possible binder. Hence, \textit{Self} can be copied upward, so that it will be satisfied by a θ-role of the matrix predicate. The relevant structure and an example are given in (81) and (82).

\[
(81) \quad \alpha \{ \text{John}_# \} \\
D \quad V \{ \text{John}, \text{John}_#, \text{Self}_# \} \\
\quad \quad \downarrow 1 \\
V \{ \text{John}, \text{John} \} \quad \beta \{ \text{John}_#, \text{Self} \} \\
\quad \quad \quad \downarrow 1 \\
D \{ \text{Self} \} \quad V \{ \text{John} \}
\]
(82) Johni expected himselfi to be the first to leave.

A second configuration in which the apparent antecedent of an anaphor is not a coargument involves PPs. In the relevant cases the anaphor is the object of a preposition that does not have an external θ-role (e.g., John talks a lot about himself). The anaphor’s function can therefore be copied out of the PP and associated with a higher predicate.

In conclusion, if the antecedent of an anaphor is a θ-role, Principle A follows from the automatic nature of function application. Of course, the classic binding theory also contains Principles B and C. These are not locality conditions and therefore do not fall out from the configurational matrix. An explanation for them must be found external to syntax proper (Reinhart 1983, Grodzinsky and Reinhart 1993, Reuland 2001).

11.4 Effects of Address Assignment

In section 9.3 we derived from Economy that the argument of an address-assigning function cannot be the source of copying. This generalization explains why a moved head cannot assign θ-roles and why A¯-movement neither feeds nor bleeds binding. It also captures the observation that A- or head-moved elements are islands (the Freezing Principle; Wexler and Culicover 1980). This is so, because Move and Head-Move, like other attributes, cannot be copied out of the argument of an address-assigning function. The Freezing Principle is illustrated for phrasal movement in (83).

(83) a. *Who did you say [[a friend of t] (that) John saw t]?
   b. *Young children John said that [[to t] you should never give matches t].

In contrast, extraction “prior to movement” is predicted to be grammatical. In particular, an A- moved constituent may contain the apparently unbound trace of NP-raising, because the thematic function that relates NP-trace to its antecedent is reconstructed. This accounts for cases of so-called remnant movement.

(84) [Painted tDP by Picasso], [the portrait] doesn’t seem to be tVP.

Since head movement also involves address assignment, moved heads are predicted to be islands. The head of a complex X0 category can excorporate, but this should be limited to base positions. This prediction is borne out by particle-verb constructions (Johnson 1991, Neeleman 1994). As (85a) shows, Icelandic allows a particle to be stranded under V-to-I movement. Stranding the particle in I, as in (85b), is impossible, however.

(85) a. [CP Í gær sendu [IP þeir t [VP ekki [VP[t upp] peningana]]]].
   ‘Yesterday, they didn’t send the money.’
   b. *[CP Í gær sendu [IP þeir [t upp] [VP ekki [VP t peningana]]]].
11.5 Intermediate Traces and Improper Movement

We have argued that a syntactic address is an island. This seems to rule out intermediate traces, which satisfy a \( \text{Move} \) function but must also introduce one. The solution to this problem lies in Accessibility. We have assumed that \( \text{Move} \) Introduction applies under containment (in the trace), but Accessibility also allows \( \text{Move} \) to be introduced in the trace’s mother, as in (86).

\[
\text{(86)} \quad \beta \{ \text{Move} \}
\]

\[
\alpha \quad t \{ \ldots, \text{address: } -, \ldots \}
\]

Let us now consider intermediate traces in more detail. We refer to the functions introduced by the intermediate and the lower trace as \( \text{Move}_I \) and \( \text{Move}_L \), respectively. The intermediate trace satisfies \( \text{Move}_L \) so that its properties are reconstructed into the lower trace. If \( \text{Move}_I \) were introduced under containment, it would reconstruct as well and therefore be unable to be satisfied by a c-commanding antecedent. But if \( \text{Move}_I \) is introduced under direct domination, it is not part of the phrase that satisfies \( \text{Move}_L \) and therefore not subject to reconstruction. Hence, \( \text{Move}_I \) can be copied upward, thereby extending the chain.

We just referred to the \( \text{Move} \) functions of the lower and intermediate traces by different names. In fact, however, they cannot be distinguished in the mother of the intermediate trace. There can only be a single \( \text{Move} \) function in that node, as shown in (87).

\[
\text{(87)} \quad \delta \{ \text{Move} \}
\]

\[
\gamma \quad \beta \{ \text{Move} \}
\]

\[
\alpha \quad t \{ \ldots, \text{address: } -, \ldots \}
\]

This is a case, then, in which a satisfied function can be copied without violating Economy. The intermediate trace is an appropriate argument for \( \text{Move} \) in \( \beta \). At the same time this trace lacks an address and must therefore itself give rise to a \( \text{Move} \) function that is to be satisfied by a category other than the intermediate trace.

Since intermediate traces satisfy a \( \text{Move} \) function, they cannot be the source of copying. This seems to be correct. Like the chain’s head in (62b), the intermediate trace in (88) cannot be the source of a \( \text{Self} \) function (the anaphor cannot be bound in the position of the intermediate trace).

\[
\text{(88)} \quad \text{Alleen zichzelf_e[ij] denkt Piet_i [t dat Jan_j t aardig vindt].}
\]

only SE-self thinks Piet that Jan nice considers

‘It is only himself that Piet thinks that Jan likes.’

NP-trace differs from \( \ddot{A} \)-trace in that the \( \theta \)-role connecting it to its antecedent is present, not as
a result of an introduction rule, but as an inherent property of the trace. It follows that this role reconstructs obligatorily if NP-trace satisfies a Move function. Hence, an A-chain cannot be extended through A-movement: that would require copying out of an argument of a Move function, contra Economy. Thus, our view of A- and NP-trace implies the ban on improper movement.

11.6 Effects of Licensing

Although licensing functions play an important role in this article, we have not made explicit what they license. A licensing function could sanction the argument that satisfies it. But why should a constituent that is well formed in isolation require licensing when embedded? A more attractive alternative takes a licensing function to sanction the structure that accommodates the constituent, rather than the constituent itself.

This raises the question how structures accommodating arguments and adjuncts differ. We assume that adjunction creates a multisegmented category, while the sister of an argument and its mother belong to different categories (Chomsky 1986a). A licensing function can then be thought of as licensing the creation of a category.

(89) Categorial Licensing

A category (as opposed to a segment) is licensed by the application of a licensing function.

Categories are sequences of nodes (segments) that have certain properties in common. In X-bar theory these were category and bar level; in the present framework segments belong to the same projection and have identical selectional properties.19

(90) Category

Let $\alpha$ be a projection of $\beta$. Let $n_{\alpha}$ be the number of unsatisfied licensing functions in $\alpha$ copied from its daughters. Let $n_{\beta}$ be the number of unsatisfied licensing functions in $\beta$. $\alpha$ and $\beta$ are segments of the same category iff $n_{\alpha}$ equals $n_{\beta}$.

What (90) says is that the arity information in the higher segment must be identical to that copied from the lower segment. The tree in (91) illustrates this. Merging an argument with a predicate creates a new category, since it affects the arity of the projected node, but merging an adjunct does not. Thus, in (91) the top two V nodes form a multisegmented category, excluding the lowest V node. As required, the creation of a category in (91) is sanctioned by the application of a licensing function.

19 The definition in (90) is a simplification, as it only considers the arity of the set of copied functions and not the actual functions that are copied. We abstract away from this here, as it does not affect the data we discuss below.
The proposal seems circular: creation of a category requires satisfaction of a licensing function, while satisfaction of a licensing function creates a category. But (90) does not refer to satisfaction. In fact, copying can create a category through an increase as well as a reduction in licensing functions. In cases of increase the empirical effects of (89) can be detected: it underlies (a) the inability of depictives to give rise to extra arguments, (b) the Adjunct Condition, and (c) the ban on NP-raising from an adjunct position.

Consider a non-selected category that introduces a licensing function. There are two scenarios: after copying, the function either is added to those of the projecting node or collapses with an identical function. These options are illustrated in (92a–b), where a depictive is combined with an unaccusative verb.

\[
\text{(92) a. } V \{Th_1, Th_2\} X \{Th_3\} \quad \text{b. } V \{Th_1\} X \{Th_2\}
\]

The number of unsatisfied licensing functions in the higher V node in (92a) exceeds that copied from the lower V node. Therefore, a category is created, which must be licensed through function application. Since this does not happen, the structure is ruled out.

In (92b) the \(\theta\)-role of the depictive is identified with the verb’s internal \(\theta\)-role. In the resulting structure the two V nodes form a multisegmented category: the number of unsatisfied licensing functions in the top node is identical to that copied from its head. Hence, no new category is created and Categorial Licensing is satisfied vacuously.

We demonstrated earlier that function identification with secondary predicates is possible (see (93a)). We have now derived that it must take place (see (93b)). This captures the observation that secondary predication is parasitic on primary \(\theta\)-role assignment (Chomsky 1986b:97–98, Williams 1994:47).

\[
\text{(93) a. } \text{John left the room angry.} \\
\text{b. } \ast \text{John seemed [that it rained] angry.}
\]

As a further consequence, secondary predicates cannot be attached higher than the highest argument in the hosting projection. In (94) ‘drunk’ must copy its \(\theta\)-role to the node labeled \(\alpha\). Since its head, \(\beta\), does not contain any unsatisfied \(\theta\)-roles, a new category is created, even though no licensing function is satisfied.
The above reasoning implies that a licensing function copied from an adjunct must be identified with a function copied from the adjunct’s sister. From this two further predictions follow: first, adjuncts should not be islands for the copying of nonlicensing functions; second, copying a \textit{Move} function from an adjunct should require identification.

The first prediction is easily confirmed. The anaphor in (95a) and the NPI in (95b) are contained in an adverbial that excludes their respective antecedents, showing that \textit{Self} and \textit{Neg} can be copied out of an adjunct.

(95) a. [The scientists], made disapproving noises [during each other’s speeches].
    b. No scientist made disapproving noises [during any of the speeches].

The second prediction relates to the observation that movement out of an adjunct is parasitic on movement in the projection hosting the adjunct. If a \textit{Move} function copied from an adjunct fails to collapse with an identical function, a new category is created, as in (96).

(96) \*\[V \{Th, \textit{Move}\} \]
    \[V \{Th\} \quad X \{\textit{Move}\} \]

The structure in (96) violates Categorial Licensing, because creation of the new category is not licensed by application of an appropriate function. Hence, extraction from adjuncts is constrained in the same way that secondary predication is, as the parallel between (97) and (93) shows.

(97) a. Which book did you file \[t \] [before reading \[e\]]?  
    b. *Which book did you sleep [before reading \[e\]]?

This analysis of adjunct islands yields the anti-c-command condition on parasitic gaps (Taraldsen 1981). If the primary gap c-commands the parasitic gap, identification is impossible. Copying of the parasitic movement function to $\alpha$ in (98) violates Categorial Licensing, as $\alpha$ does not contain the function introduced by the primary gap.

(98) *Who did you explain \[t \{\alpha \textit{met you} \} \] [before recognizing \[e\]]?  

As explained, \textit{Move} Introduction can apply under immediate domination. This not only affects intermediate traces, but also explains apparent violations of Categorial Licensing. Consider (99), which contains a trace left behind by adjunct movement.

(99) \[\alpha \{\textit{Move}\} \]
    \[\alpha \quad t \{\ldots, \text{address: \texttt{-}, \ldots}\} \]

Categories are defined in terms of the number of functions copied to a node, as compared with
the number of functions copied from its head. If these diverge, the higher node forms a new
category. In (99) the Move function is not copied from the trace but directly introduced in its
mother. This makes it irrelevant to categoryhood, with the effect that the lower and the higher
instances of $\alpha$ form a multisegmented category. Consequently, Categorial Licensing does not
come into play and adjunct extraction—as opposed to extraction from an adjunct—is allowed
(e.g., *When did John say [that he left $t$]*).

NP-trace differs from $\mathbb{A}$- and $X^0$-trace in that the function connecting it to its antecedent is
an inherent property of this element. Hence, NP-raising from an adjoined position should violate
Categorial Licensing. Indeed, although *yesterday* can be a subject, raising as in (100) is ruled
out.

(100) *Yesterday seemed $t$ that John left.

12 Summary

Our theory of grammatical dependencies is based on four central assumptions:

1. *Inclusiveness*
   The syntactic properties of a nonterminal node are fully recoverable from the structure
   it dominates; the syntactic properties of a terminal node are fully recoverable through
   mapping procedures.

2. *Accessibility*
   Relations between nodes require immediate domination.

3. *Distinctness*
   The syntax interprets attributes of a node that cannot be distinguished as one and the
   same.

4. *Economy*
   Information in a node must be minimally motivated in its immediate environment (as
   defined by Accessibility).

These assumptions have forced revisions of how grammatical dependencies are encoded. In develop-
oping our proposals, we have assumed these additional conditions and rules:

1. *Exclusivity*
   Consider a structure $[\gamma \alpha \beta]$. Let $n_\alpha$, $n_\beta$, and $n_\gamma$ be the number of unsatisfied licensing
   functions in $\alpha$, $\beta$, and $\gamma$, respectively. If a licensing function is satisfied in $\gamma$, then $n_\gamma$
   may not be less than $n_\alpha + n_\beta - 1$.

2. *Categorial Licensing*
   A category (as opposed to a segment) is licensed by the application of a licensing function.

3. *Move Introduction*
   $[\text{Address: } - ] \Rightarrow \{ \ldots, \text{Move}, \ldots \}$

4. *Hd-Move Introduction*
   $[\text{Address: } - ] \Rightarrow \{ \ldots, \text{Hd-Move}, \ldots \}$
The resulting syntax subsumes the configurational matrix (points 1 to 5 below). It also gives rise to the empirical generalizations in points 6 to 30, some of which overlap.

1. If \( \alpha \) is grammatically dependent on \( \beta \), \( \beta \) c-commands \( \alpha \).
2. A dependent can enter into the same external dependency only once.
3. A dependent can enter into the same internal dependency more than once.
4. An antecedent can enter into the same grammatical dependency more than once.
5. Dependent elements obligatorily take an antecedent.
6. An argument cannot receive more than one primary \( \theta \)-role, but it can receive additional secondary \( \theta \)-roles.
7. Predicates can function as prenominal modifiers; elements without a \( \theta \)-role, such as impersonal passive participles, cannot.
8. There is no NP-raising to positions to which a primary \( \theta \)-role is assigned, but NP-raising is allowed to positions that satisfy a secondary \( \theta \)-role.
9. There is no NP-raising to object position.
10. There is no NP-raising from an adjunct position.
11. \( \bar{A} \)-chains cannot be extended through A-movement.
12. Predicates can be coordinated with passive and unaccusative VPs.
13. There is no syntactic reconstruction with NP-raising.
14. There is obligatory syntactic reconstruction with \( X^0 \)- and \( \bar{A} \)-movement.
15. Anaphors can be bound in a derived subject position.
16. There is no selection from positions derived by \( \bar{A} \)- or head movement.
17. Adjuncts are islands for movement, but they can contain parasitic gaps.
18. Adjuncts themselves can move.
19. A parasitic gap cannot be c-commanded by the trace whose antecedent it shares.
20. There is no primary predication from adjuncts, but adjuncts can be secondary predicates.
21. Adjuncts are transparent for binding and the licensing of NPIs.
22. There is no NP-raising across a subject.
23. There is no predication across a subject.
24. Clauses introduced by \( w/h \)-elements are islands for \( w/h \)-movement.
25. There can be no head movement across a head.
26. Excorporation of nonheads is excluded; excorporation of heads is allowed in base positions.
27. C-selection is restricted to internal arguments.
28. Elements that have undergone \( \bar{A} \)- or head movement cannot enter into grammatical dependencies, except as intermediate traces.
29. Elements that have undergone NP-raising can enter into new grammatical dependencies.
30. A-movement can give rise to remnant \( \bar{A} \)-movement.

References


Department of Phonetics and Linguistics
University College London
Gower Street
London WC1E 6BT
United Kingdom

ad@ling.ucl.ac.uk
hans@ling.ucl.ac.uk