Deriving Narrow Syntax through constraints on information structure: a parallel between linguistic models of displacement and database theory

Mark de Vos

Abstract

This paper is the 'Accepted Authors Manuscript' version of a paper published in Lingua:

De Vos, M. 2008. Deriving Narrow Syntax through constraints on information structure: a parallel between linguistic models of displacement and database theory. Lingua 118 (2008) 1864–1899. [DOI: http://dx.doi.org/10.1016/j.lingua.2008.05.002].

It is provided on this webserver as a draft copy for scholarly purposes. Please do not cite this version but rather reference the original. Comments are welcome as always.

This paper presents a research program for normalization-driven syntax. It takes the Minimalist research agenda as a starting point (Chomsky 1995. *The Minimalist Program*. MIT Press, Cambridge MA.) and explores the question of how the CI interface determines syntactic operations. The proposal provides specific content to the notion of bare output conditions and the nature of the CI interface. It does so by drawing on the tools provided by Relational Theory, a branch of set-theoretic mathematics, and Database Theory, a branch of computer science. It is demonstrated that core components of Narrow Syntax (phrase structure, selection and AGREE) are all definable in terms of Relational Theory. Then, it is shown that the process of relation optimization, or normalization, can derive chain formation. The article concludes with two speculations on the implementation of phases within a normalization-driven grammar and the implications of such a system for the learnability of the lexicon.

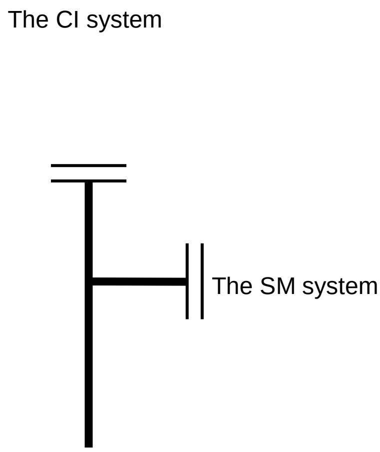
Department of English Language and Linguistics, Rhodes University, PO Box 94, 6140 Grahamstown, South Africa

m.devos@ru.ac.za

1 Desiderata for perfect architecture

Over the past decade, the Minimalist Program (Chomsky 1993; 1995b; 1998; 2000; 2001a;b; 2002) has stimulated a lot of research into the architecture of the human language faculty. In particular, research within the Minimalist Program has shown that many core properties of Narrow Syntax are determined by bare output conditions – these being that the syntactic system must interface with a number of other modules, including the Conceptual-intentional (CI) and the Sensory-motor (SM) systems.[1] This is illustrated in figure 1.

Figure 1: Syntax interfaces with the CI module(s)



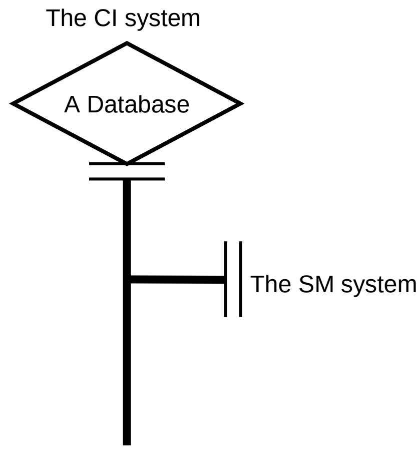
I will further assume the null hypothesis that there is a one-to-one, meaning-preserving mapping between syntactic relationships and LF representations.[2]

While much is known about syntax, considerably less is known about the interfaces, especially the CI interface. However this is not to say that the abstract properties of the CI interface are not known. Various proposals have been made (Chomsky 2001b, Epstein 1999, Uriagereka 1999). Common with all of these is the idea that the CI system is structured and orderly, possibly fed by multiple Spell Out. On the understanding that a database is a set with regular structure and knowing that knowledge within the mind is structured, it is a possible hypothesis that some part of the CI module/s must interface with an organized system for knowledge representation (informally, a database: a set with regular structure). This is illustrated in figure 2 on the next page.[3] Assuming that CI determines Narrow Syntax and that there is a one-to-one mapping between the two, Narrow Syntax produces representations that are interpretable and unambiguous to the CI system. And thus, within the Minimalist Program, it is feasible to query to what extent the properties of Narrow Syntax are determined by the properties of the database with which it interfaces.

This can also be framed in terms of a hypothesis (1a) reminiscent of Chomsky's Strong Minimalist Hypothesis.

(1) Syntactic representations are determined by:

Figure 2: Narrow Syntax interfaces with a database within the CI module/s



a. the properties of information structure as formalized by relational databases

b. and by further properties of the PF interface.

This paper focusses on the way in which narrow syntax may be organized according to principles of relational databases. Although (1b) no doubt plays a role in determining syntactic representations, it is beyond the scope of this paper. Note that I am not arguing that it is a logical necessity that Narrow Syntax is determined in this way – it is merely a hypothesis ultimately derived from the Strong Minimalist Hypothesis. After all, there are potentially any number of mappings between CI and Narrow Syntax. Determining which mappings are actually instantiated is a matter for empirical research. The central concern of this paper is to investigate the extent to which a particular type of conception of CI – namely as a relational system – affects Narrow Syntactic representations. To the extent that the proposal is successful in analyzing syntactic data, it will also provide evidence for both the mapping and the database proposal.

1.1 Structure of this paper

The structure of this paper is as follows. Since this paper is based largely on Relational Theory and the known properties of databases, I devote considerable space to outlining exactly what databases are and how Relational Theory approaches them. Section (2) outlines the properties of databases within a relational model. Concepts introduced include the central notion of Functional Dependencies and also Normal Forms as well-formedness conditions on relational databases. This is illustrated with concrete, non-linguistic examples.

Section (3) explores these same notions from a syntactic perspective using linguistic examples. It is shown that syntactic constructs such as Phrase Structure, Selection and AGREE can be represented as partially-ordered sets or functional dependencies. This does not mean that all partially-ordered sets are necessarily equivalent to well-formed syntactic trees![4] However, it is necessary to frame syntactic relations in set-theoretic terms in order for the comparison with Relational Theory to be made.

Section (4) can be considered the core of this paper as it takes this parallel one step further, showing that syntactic trees, including movement chains, can be captured by *normalized functional dependencies*. This is the core claim of this paper.

A normalization-driven grammar has implications for the architecture of the grammar. This is discussed in section (5) where the effects of such a model on binarity, movement and other aspects of syntactic theory are briefly discussed.

To the extent that the argument of this article is successful, the resulting normalization-driven grammar could have broader implications. The article ends with a bold speculation about the implementation of phases within such a model (section 6.1) and the possible extension of a normalization-driven syntax to lexical relationships (section 6.2).

1.1.1 Limitations of this paper

This paper cannot be exhaustive; it is an ambitious attempt to illustrate a parallel between Relational Theory and Linguistic Theory. Due to the amount of material that must be covered, this paper limits itself to a theoretical approach, leaving deeper empirical effects to one side. However, to the extent that this parallel is justified, there are probably grounds to explore the research program further. In addition, the paper largely limits itself to the contribution of the CI interface (1a), leaving the impact of the PF interface (1b) to one side.

2 The properties of databases

As a starting point, it is pertinent to ask what the general principles governing well-formedness in databases are and whether these same principles are operational in Narrow Syntax. Some of these questions have already been partly addressed by the Relational model of information representation (Codd 1970; 1983), couched in formal set theory, which was developed outside a properly linguistic or even cognitive context. A database is a set with regular structure. In the 1960s, research into the structure of databases resulted in several models: the flat model, the network model and the hierarchical model. Finally, the Relational model, based on set theory, was developed for the modelling of databases. Codd (1970) demonstrated that the relational model was significantly superior to other methods in a number of respects. The relational model is the most widely used today.

2.1 The relational model

Any text file can list random information, but knowledge requires a database: an information set with a regular structure. As a minimum, the representation of knowledge requires the formalization of a relationship between two entities. This may be done through combining them into a relation (i.e. a table or partially ordered set) where one item determines the other. The following trivial example represents the knowledge that *John* is an *Office worker* using three, formally equivalent representations.

(2) a.

John	Office

b. (JOHN, OFFICE)

c. {JOHN, {JOHN, OFFICE}}

In relational models, data are stored in tables (relations which are equivalent to partially ordered sets). A column of a table/relation is called an attribute and a row is called a tuple or record. Each table/relation should have one or more attributes as *primary key* which determines all the other attributes.

(3) a. **Candidate key** An attribute/column (or combination of attributes) with minimum cardinality that uniquely identifies a tuple/row in a given relation (Dutka and Hanson 1989).

b. **Primary key:** A candidate key that is used as the unique identifier of a tuple/row.

Informally, this means that if one knows the primary key, then one can find a unique tuple/record in the table/relation.

To use a concrete example, suppose that John, Peter and Mary are all office workers for a company and all receive the same bonus, namely $200. In addition, their physical location within the corporation depends on what kinds of job they do. Office workers are located on the 2nd floor, managers on the third floor. This information might be represented in a table/relation as follows.

(4)

EMPLOYEE-ID	JOB	BONUS	LOCATION
John	Office	200	2nd Floor
Peter	Office	200	2nd Floor
Mary	Office	200	2nd Floor
Sarah	Manager	500	3rd Floor

(5) (EMPLOYEE-ID, JOB, BONUS, LOCATION)

In this table/relation, the primary key is the attribute EMPLOYEE-ID. This means that if one knows the EMPLOYEE-ID then one can find any particular record in the database. For instance, if one knows the EMPLOYEE-ID *John*, then one can also find out that *John* is an *Office Worker*, earns *$200* and works on the *second floor*. The inverse is not true: if one knows that somebody earns *$200*, then you cannot infer that the person in question is *John*. This means that BONUS is not the primary key.

2.2 Functional dependencies

In Relational Theory, the relationships in (4) are depicted using arrows to indicate functional dependencies.

(6) a. EMPLOYEE-ID $\longrightarrow$ JOB

b. JOB $\longrightarrow$ BONUS

c. JOB $\longrightarrow$ LOCATION

Functional dependencies are a theoretical notion in Relational Theory (Codd 1970) corresponding to partially ordered sets and which are, informally, directions of dependency within a set.

(7) a. **Functional Dependency:** X functionally determines Y if the value of X determines the value of Y (i.e. $X \to Y$).[5]

b. **Value:** The value is the information content of a particular tuple within a table/relation.

To illustrate this definition, consider the relationships present in table 4 on the preceding page. Values for this relation include *John*, *Office* and *2nd Floor*.

The EMPLOYEE-ID determines the JOB that is done by each employee (6a). Practically speaking, if we know somebody's EMPLOYEE-ID then we also know what JOB they do. This relation is not reversible since, even if we know what the JOB of somebody is, we cannot immediately determine what their EMPLOYEE-ID is.

The JOB determines what the BONUS is (6b). If one knows what the JOB of somebody is, then their bonuses can be determined. JOB also determines LOCATION since it was specified that office workers are on the second floor (6c). Thus, knowing that somebody is an office worker is sufficient to infer that they also work on the second floor.

2.3 Normal Forms and normalization: well-formedness conditions on relational databases

A major challenge for any system of knowledge representation is that it must have the ability to be enlarged, modified or updated without destroying the integrity or consistency of the data.[6]

A database such as the one in 4 on the page before faces some challenges when it comes to being updated. The reason for this is that the relations EMPLOYEE-ID → JOB and JOB → BONUS together constitute a transitive dependency.

Imagine that all the office workers were granted a pay rise to, say, $300. Then, in order to keep the database up to date, every single record of every single employee attribute would have to be updated. In table 4 on the preceding page this would mean that $200 would have to be updated three times; in a larger database with potentially thousands of employees, the computational cost is extreme. The table holds redundant information which takes additional resources to update and increases the chances of an error occurring.

In order to avoid these kinds of problems, a database must be optimized; it must obey several mathematically well-defined constraints on data representation. These constraints are called *Normal Forms* (NF) and the process of enforcing Normal Forms is called *normalization*.

There are several mathematically well-defined levels of normalization.[7] Normal Forms are incremental: if a set is in 2NF then it is automatically in 1NF. If a set is in 3NF, it is also in 2NF and 1NF, etc. Normal Forms crucially rely on Functional Dependencies and correspond to the following conditions on knowledge representations (which will be explained shortly).[8]

(8) For any relation, a particular Normal Form is met when the following conditions apply:

1NF: attributes are atomic and non-repeating,

2NF: the relation is in 1NF and non-key attributes are functionally determined by the entire primary key,

3NF/Boyce-Codd NF: the relation is in 2NF and all non-key attributes are functionally dependent on a candidate key

4NF: the relation is in 3NF/BCNF and there are no multi-valued functional dependencies.

The process of enforcing these rules and thereby optimizing the database is called normalization (Codd 1970; 1983). Normalization is (i) a formal framework for analysing relations and (ii) a set of tests that can be applied to any relation schema so that the entire database can be optimized to ensure the integrity and consistency of the data under modification. If a test fails then the offending relation schema must be decomposed into smaller relations which do pass the tests. Normalization rules are thus a necessary set of independently motivated rules for all relational databases that ensure coherence of data and efficient computing when the database is updated.[9] These Normal Forms are discussed below with reference to concrete examples.

2.3.1 1NF

Relations in 1NF only have atomic attributes. There are three ways in which data may not be atomic. (a) An attribute could contain unparsed text (b) the attribute may be a combination of domains and (c) an attribute may contain a repeating group (where the same attribute name is used to refer to multiple instances of data of the same type). In the following example, the BONUS MONTHS attribute is a combination of various types of information. It is not parsed. This is not a problem if it is treated as an atomic string, but prevents the information from being processed separately.

(9)

EMPLOYEE-ID	BONUS MONTHS
Peter	Jan, Aug
Mary	Jan, June, Oct

A relation in 1NF does not have repeated attributes. In the following example, the attribute TELEPHONE NO. is used more than once to reflect the fact that an office worker may be accessed by several telephones. The attribute TELEPHONE NO is repeated.

(10)

EMPLOYEE-ID	JOB	TELEPHONE NO.	TELEPHONE NO.
Mary	Office	0642252509	0715129069
John	Office	556469	–

Now consider the database in 4 on page 5, repeated here as 11.

(11)

EMPLOYEE-ID	JOB	BONUS	LOCATION
John	Office	200	2nd Floor
Peter	Office	200	2nd Floor
Mary	Office	200	2nd Floor
Sarah	Manager	500	3rd Floor
Bill	Programmer	300	2nd Floor

(12) (EMPLOYEE-ID, JOB, BONUS, LOCATION)

This relation is in 1NF: all the data is atomic and there are no repeated attributes.

2.3.2 2NF

A relation is in 2NF if there is a mapping between the primary key and all other information in the relation. More formally stated:

(13) **2NF:** a relation is in 2NF if (i) it is in 1NF and (ii) every non-key attribute is fully functionally dependent on the primary key (Dutka and Hanson 1989:26).

Relations which are not in 2NF typically have attributes which are completely unrelated to the primary key. The relation 11 on the page before is in 2NF because it is in 1NF and all the data are ultimately dependent on the primary key – there is some mapping between the primary key, EMPLOYEE-ID, and everything else. So, for instance, if one knows the EMPLOYEE-ID for any worker, then one can also determine their JOB, LOCATION and BONUS.

Note that 2NF holds even if a transitive dependency exists in the relation, as in 11 on the preceding page, where LOCATION is dependent on JOB and JOB is dependent on EMPLOYEE-ID; LOCATION is apparently not directly dependent on EMPLOYEE-ID. The definition in (13) does not require *direct* functional dependence – merely *full* functional dependence. In other words, there is some function which relates EMPLOYEE-ID to LOCATION; the fact that the relationship between them is mediated is immaterial. In terms of Armstrong's axioms (Armstrong 1974), Functional Dependencies are transitive.

(14) **Transitivity:** if aRb and bRc then aRc

Thus, for any transitive dependency, if A $\rightarrow$ B and B $\rightarrow$ C then A $\rightarrow$ C. Thus, the table in 11 on the page before is in 2NF because all attributes are ultimately dependent on the primary key.

2.3.3 3NF/BCNF

Although the relation 11 on the preceding page is in 2NF, there are still redundancies which need to be normalized, as described in section 2.2. There are transitive dependencies:

(15) a. EMPLOYEE-ID $\rightarrow$ JOB $\rightarrow$ LOCATION

b. EMPLOYEE-ID $\rightarrow$ JOB $\rightarrow$ BONUS

Imagine that the company were to hire five new office workers. Then, the information that Office workers earn $200 would have to be rewritten five times. This level of redundancy increases the computational load during updating. Similarly, imagine that all the office workers were fired and their records deleted from the database. This would mean that the fact that office workers earn $200 would also be lost from the system.

The third level of normalization we will be discussing, Boyce-Codd Normal Form (BCNF), is largely concerned with removing transitive dependencies.[10]

(16) **Boyce-Codd Normal Form:** A relation is in BCNF if (i) it is in 2NF and (ii) non-key attributes are all dependent on a candidate key.

To avoid the problems associated with transitive dependencies it would make more sense to create two separate tables where the key of the first table is EMPLOYEE-ID and the key of the second is JOB.

(17)

EMPLOYEE-ID	JOB
John	Office
Peter	Office
Mary	Office
Sarah	Manager
Bill	Programmer

JOB	BONUS	LOCATION
Office	200	2nd Floor
Manager	500	3rd Floor
Programmer	300	2nd Floor

(18) (EMPLOYEE-ID, JOB) (JOB, BONUS, LOCATION)

Now, if the salaries are increased, only one attribute in one relation needs to be changed. This test was based on the addition of information to the database. A similar test can be done based on the notion of data elimination. Consider the original relation. Assume that John, Peter and Mary were all fired and their records removed from the database. Important information would be lost, namely that office workers always receive $200. Thus, when a new set of employees are added to the database, it fails to represent how much money they are supposed to get. Now consider the second set of tables in (17). Even if John, Peter and Mary are all fired and their data removed, the BONUS of office workers is not removed from the system as illustrated in (19).

(19)

JOB	BONUS	LOCATION
Office	200	2nd Floor
Manager	500	3rd Floor
Programmer	300	2nd Floor

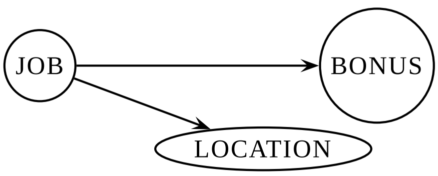
(20) (JOB, BONUS, LOCATION)

2.3.4 4NF

However, the table in 19 is still not problem free. In the unlikely event of the bonus for managers being scrapped, the entire second row would have to be deleted. Then the information that managers also work on the third floor would be lost. In other words, there is an ambiguity in this table. Alternatively, imagine managers worked on a variety of different floors. Then, in order to update the database, for every new location, the bonus would have to be specified redundantly.

The problem with table 19 is that it fails to represent unambiguously a multi-valued dependency. JOB determines both BONUS and LOCATION, but neither is related to the other (a multi-valued dependency).

(21)



Consequently, in order to optimize the database, it is necessary to split the relation in 19 into two simpler ones.

(22)

JOB	BONUS
Office	200
Manager	500
Programmer	300

JOB	LOCATION
Office	2nd Floor
Manager	3rd Floor
Programmer	2nd Floor

(23) (JOB, BONUS) (JOB, LOCATION)

The full set of normalized relations is illustrated below. There are no multi-valued dependencies; each relation is at least in 4NF. In short, these relations represent the optimal representation of this data, given the constraints imposed by modification and deletion.

(24) (EMPLOYEE-ID, JOB) (JOB, BONUS) (JOB, LOCATION)

2.4 Interim conclusions

Section 2 has described the properties of databases. It has been claimed that (i) relational databases are defined over partially ordered sets and (ii) they are subject to well-defined well-formedness conditions which are directly related to preserving the integrity of the data. Importantly, none of these claims are controversial within Relational Theory.

3 Extending normalization to syntactic representations

The following section argues that Relational Theory and normalization can be fruitfully applied to syntactic representations. The first step in the argument is to show that core components of Narrow Syntax (i.e. phrase structure, selection and AGREE) can be defined in terms of Functional dependencies, or partially-ordered sets (section 3.1). The second step is to show that normalization, as applied to syntactic relationships yields well-formed syntactic representations (section 4).

Partially ordered sets have the form {A,{A,B}}. This is also often written as (A,B). A partially ordered set is reflexive, transitive and antisymmetric. If any relation *R* defined over *a* and *b* has these properties, then *a* and *b* constitute a partially ordered set (Devlin 1993, Halmos 1960).[11]

(25) a. **Reflexivity:** aRa
b. **Transitivity:** if aRb and bRc then aRc
c. **Antisymmetry:** if aRb and bRa then $a = b$

Within Relational Theory, partially ordered sets can also be regarded as Functional Dependencies (Codd 1970). This means that partially ordered sets are a useful notation with which to compare Relational Theory and Syntactic Theory. Drawing on the definition of Functional Dependencies provided by Dutka and Hanson (1989), I define Functional Dependencies in linguistic terms as follows.

(26) a. **Functional dependency:** Let A and B represent sets of syntactic features (trivially including sets of just one feature). A functionally determines B if the value of A determines the value of B (i.e. A $\rightarrow$ B).
b. **Value:** Let the value of A and B be the value of features (e.g. categorial features $\pm$N, $\pm$V; formal features uϕ, ϕ; semantic features $\pm$agent etc).

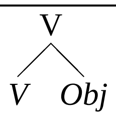
Of course, dependency is a central component of many linguistic theories. The definition in (26) merely formalizes this intuition in a principled way. Consider how Functional Dependencies apply to linguistic notions such as phrase structure, selection and agreement.

3.1 *Mapping Phrase structures to Functional Dependencies*

Phrase structures created by MERGE are partially ordered sets (Chomsky 1995a, Uriagereka 1999). Given two syntactic elements, A and B, when they undergo MERGE, a partially ordered set {A,{A,B}} (Chomsky 1995b) is formed.[12]

Mathematically speaking, this can be represented by the functional dependency: A $\rightarrow$ B.[13] Thus, just as it is possible to represent a phrase-structural relationship, say V merged with an Object DP, as a tree, it is also possible to represent it as a relation/table or as a set. Doing so is a representational device but nevertheless illuminating.

(27)

SET	≡ TREE/GRAPH	≡ FUNCTIONAL DEPENDENCY
{**V**,{V, Obj}} ≡		≡ V $\rightarrow$ Obj

Drawing on my assumptions in section (1), I take it as given that there is a one-to-one mapping from set-theoretic phrase structure to functional dependencies.

3.1.1 *Heuristics for combining phrase markers into trees*

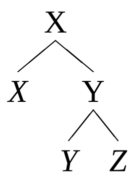
In this paper, I will repeatedly refer to (normalized) sets rather than (graphic) phrase-structure markers. For example (X,Z) can easily be captured from the relational representations (X,Y) and (Y,Z) by the transitivity rule.

However, interested readers may be curious about how the two representations map onto each other. Given that both normalized sets and (bare) phrase structure are expressed in terms of partially ordered sets they can be translated fairly easily by substituting one set into another, as illustrated in (28).[14] The graphic representation in (28c) encodes the original ordered pairs.

(28) Substitution of Y yields:

a. {X, {X, Y}} and {Y, {Y, Z}}

b. {X, {X, {Y, {Y, Z}}}}

c. 

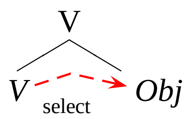
Note that these heuristics are not derivational statements.[15] They are merely provided to assist the reader in converting the set notation used in this paper into syntactic representations which are easier to visualize.

3.2 *Selection*

The ability of a syntactic head to select a complement is determined by a subcategorization feature upon the selecting head.[16] Thus, by an inspection of a particular feature, one can ascertain what kind of element it will merge with.

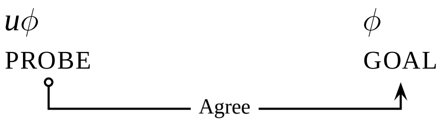
To illustrate this, one might consider the head V-transitive which has a subcategorization property for an object. Thus, V is deterministically related to the object: V $\longrightarrow$ Obj. Such a relationship can be expressed as a set-theoretic relation equivalent to a Functional Dependency or a tree structure. In (29b), the dotted line informally represents the Functional Dependency.

(29) a. V $\longrightarrow$ Obj

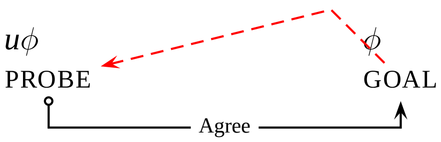
b. 

3.3 *Agreement*

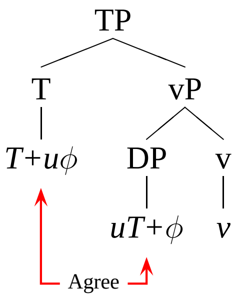
Agreement can also be considered an instantiation of a Functional Dependency. AGREE is an asymmetric, binary relationship between a probe (uϕ) and a goal (ϕ) (Chomsky 1998; 2000).

(30) 

For example if this relation instantiated NUMBER agreement, then, the probe might have an initial value of uϕ while the goal could have a value of +PLURAL. In an approach where AGREE is feature valuation (Chomsky 2001a), after AGREE has occurred the value of both the probe and the goal will then be +PLURAL. The result is that the value of the goal determines the value of the probe: a Functional Dependency.[17] Thus, AGREE creates a relation of the form (F, uF) where the value of F ultimately determines that of uF. In the following diagram, the Functional Dependency is informally illustrated with a dotted line.[18]

(31) 

This can be demonstrated in more detail with a typical agreement relationship between a DP in Spec *v*P and T. Assume that Nominative case is a reflex of uninterpretable T on the DP (Pesetsky and Torrego 2001). The DP has ϕ features such as NUMBER to be checked against the corresponding uninterpretable features on T. On the other hand, uninterpretable T on the DP needs to be checked by the T head. If one considers these relationships carefully, then it can be seen that they are Functional Dependencies as shown in (32).

(32) 

a. T $\longrightarrow$ uT
b. $\phi \longrightarrow$ uϕ

For instance, the value of the ϕ features on the DP (e.g. PLURAL) will determine the value of the corresponding features which are ultimately realized morphologically on T. Similarly, if one takes Case to be mnemonic for uninterpretable Tense (Pesetsky and Torrego 2001), then the value of the T feature will determine the resulting case on the DP. Thus, AGREE is an instantiation of Functional Dependency: $\phi \rightarrow u\phi$.

3.4 Interim conclusions

Phrase structure, selection and AGREE can all be represented as Functional Dependencies. Functional Dependencies are partially ordered sets. This equivalence between relational sets and trees suggests the following principle.

(33) **Relational Equivalence:** Syntactic, structural relationships are set-theoretic and can thus be represented as set-theoretic relations in Relational Theory.

With this in mind, let us explore a corollary to Relational Equivalence (33) which is expressed in (34).

(34) **Corollary to Relational Equivalence:** Since relations can and must be normalized, syntactic phrase structure relations must also be normalized.

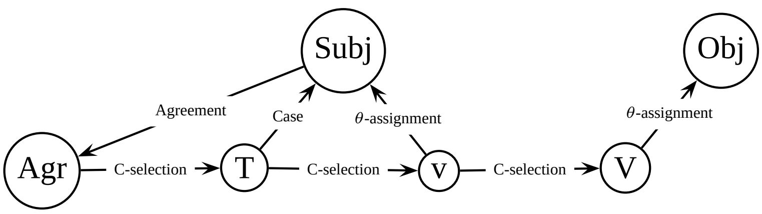
The following section explores the implications of Relational Equivalence (33) and its corollary (34) for the representation of linguistic relationships. It will demonstrate that normalization techniques can be fruitfully applied to the set-theoretic representations created by Narrow Syntax with interesting results.

4 Grammatical relationships and normalization

As a starting point, it is uncontroversial that grammatical relationships such as selection, predication, case assignment and agreement exist. Selectional properties of the functional hierarchy are thus taken as a primitive. There are also relationships of agreement, predication etc. which exist between various categories that fulfill functions of subjects, predicates, objects etc.[19] These relationships are part of every syntactic tree. They can all be reduced to set-theoretic terms and that is what is of primary interest at the moment. Thus a sentence like *Sarah eats oranges* could be broken down into relationships like the following ones.

(35) a. Sarah eats oranges

b.



T assigns case to the subject which *v* introduces as an external argument (a multi-valued dependency). The subject checks agreement features on Agr and Agr C-selects T and T C-selects *v* which also C-selects V (a transitive dependency). V also L-selects the object, assigning a theta-role and is thus potentially part of the Subj-Agr-T-*v*-V transitive dependency. According to **Relational Equivalence** (33) section (3.1), all these relationships can be represented as set-theoretic objects.[20]

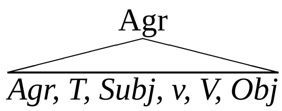
4.1 1NF

These relations can be grouped in a complex relation in 1NF which can be visualized either as a relation (Agr, T, Subj, v, V, Obj) (36a) or as a n-ary tree (36b).

(36) a.

Agr	T	Subj	v	V	Obj

b.

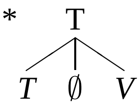


This relation is in 1NF because all elements are atomic, parsed and non-repeating. In fact, linguistic tree structures are trivially in 1NF;[21] tree structures and relations are governed by the same general principles. To illustrate this, the relation in (37) is not in 1NF because an attribute is not filled in. The attributes are therefore not atomic.

(37) a.

T	–	V

b. *



It is trivially true that a linguistic tree cannot have a node that is radically unfilled; an absent node cannot select or be selected. In this respect, relations and trees are governed by the same constraints: both must be in 1NF.[22]

4.2 2NF

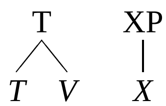
Second normal form requires that all attributes are related to the primary key. The relation in (36) is also in 2NF because all the attributes are related to each other.

Before continuing, it is worth while to consider a scenario where a relation is in 1NF but not in 2NF. An imaginary relation like (38) would not be in 2NF if, for instance, T $\rightarrow$ V but XP is isolated and is not functionally determined by anything. Note that this is *not* evident from the table structure itself – the notation has let us down. Although (38) bears a superficial resemblance to (36), they are actually quite different given the assumption that XP is not functionally determined; (38) has an XP which is not related to anything while the constituents of (36) all relate to each other.[23] This is equivalent to a linguistic tree where a constituent is not dominated by any other constituent. It is, in effect, in its own tree. Thus, it is a trivial property of linguistic trees that they be at least in 2NF.

(38) a.

T	V	XP

b.

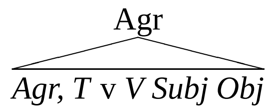


Returning to the main discussion of 2NF, it is clear that 2NF by itself is insufficient to account for linguistic effects. For instance, 2NF does not prevent the creation of n-ary trees. The relation in (39) is in 2NF and corresponds to a non-binary tree (40).[24]

(39)

Agr	T	*v*	V	Subj	Obj

(40)



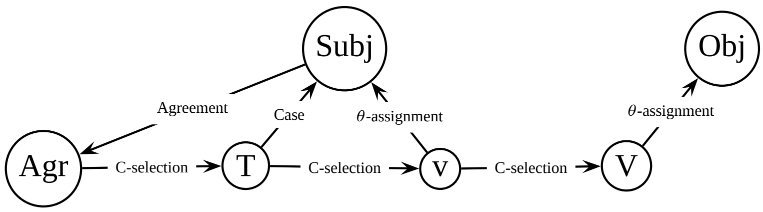
From this it follows that while not all 2NF relations are well-formed syntactic trees, all well-formed syntactic trees are at least in 2NF.

4.3 3NF/BCNF

It has been shown that all linguistic trees must be in 1NF and 2NF in order to be well-formed. This is quite significant in itself since there is no a priori reason for this to be the case. Now consider the full set of relations described in 35 on page 13 and repeated here as 41.

(41) a. Sarah eats oranges

b.



Although they can be represented as a table of relations, for the sake of convenience let us represent them as bracketed relations in a numeration. Note that the ϕ features on D are a subset of the Subj feature bundle. Let us call this Stage 1.

(42)

Stage 1
(Agr, T, *v*, Subj[ϕ], V, Obj)

The relation is in 1NF because all attributes are atomic and parsed. It is in 2NF because all attributes are ultimately fully functionally determined by the primary key. However, the presence of transitive dependencies (Agr $\rightarrow$T $\rightarrow$ v $\rightarrow$ V $\rightarrow$ Obj) means that the relation is not in 3NF. As discussed earlier, there are good reasons why a database should be in 3NF. Linguists will also argue that there are good reasons why trees cannot always be n-ary (Chomsky 1995a, Kayne 1984; 1994). In other words, there is agreement from both linguistic and database perspectives that the representations in (42) cannot remain as they are. The solution is to break up the large relation into a number of small ones in which there are no transitive dependencies.

The most natural way to do this is to place all attributes of the original table that prevent it from being in 2NF in a new table.

(43) **Normalize PS:** Make new relations of only those data attributes which cause the relations in which they occur to fail a particular level of normalization. Leave all non-offending attributes in the original table. Tables which pass a particular level of normalization are not affected.

This procedure is represented in (44). The double strikeout indicates that attributes which caused the relation to fail the normalization constraint have been deleted and placed into new tables.

(44)

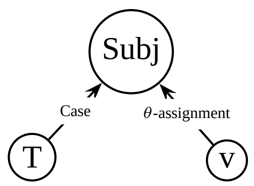
Stage 2:	Stage 1
(T,*v*, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(	~~Agr, T, *v*, Subj,~~ V, Obj)

Note that the original relations present are intact (i.e. they have been neither modified nor destroyed). These are illustrated graphically below (refer to the illustration in (35) for comparison). The transitive C-selection relationships between Agr, T, *v* and V are decomposed into two smaller relations: (Agr,T), (T,*v*) and (*v*,V) respectively. In addition the agreement relationship between Subj and Agr is represented in (Subj[ϕ],Agr).[25]

(45)
a. (Agr,T) Agr → T — C-selection
b. (T,*v*) T → v — C-selection
c. (*v*,V) v → V — C-selection
d. (Subj[ϕ], Agr) Subj → Agr — Agreement

Similarly, the various relationships of Case, Agreement and Theta-assignment that exist between T, Subj and *v* are contained in the following relation (T, *v*, Subj) in (46). Importantly, as illustrated by the graphic representation, this particular relation does not include any transitive dependencies; the relations between T and *v* and Subj and Agr are *not* included in this particular relation (46) since they are expressed by the relations (T,*v*) and (Subj[ϕ],Agr) respectively.

(46)
a. (T, *v*, Subj)
b.



4.4 4NF

Fourth normal form (4NF) involves removing multi-valued (i.e. one-to-many and many-to-one) dependencies. Most of the relations in (44) are in 4NF. However, the leftmost table, (T,*v*, Subj), contains a many-to-one multi-valued dependency; it is not in 4NF. Normalization of this structure deletes the offending elements from the table and creates new tables that are both in 4NF. In addition, all the remaining relations in (47) are at least in 4NF and need not be normalized further.

(47)

Stage 3:	Stage 2:	Stage 1
(T, Subj):(	~~T~~,*v*, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(	~~Agr, T, *v*, Subj,~~ V, Obj)

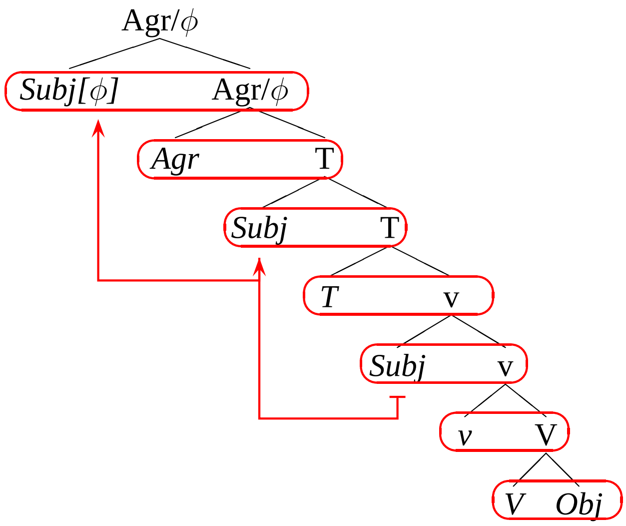
The ‘cleaned up’ version of these relations is in (48a). For the sake of visualization, these relations can be mapped to a partially ordered set in (48b) and composed into a single phrase structure (49) by set-union.[26]

(48) a. (T, Subj) (v, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V) (V, Obj)

b. {Subj[ϕ], {Agr[ϕ], {T, {Subj, {T, {T, {v, {Subj, {v, {v,{V,{V,Obj}}}}}}}}}}}

The ovals informally indicate that all the original relationships have been preserved; the arrow informally indicates that a movement chain has been created.

(49)



The resulting structure bears several similarities to a syntactic tree derived by standard means: (i) there is a Subject-chain and (ii) the phrase structure displays the effect of binarity. This is evidence that the parallel between normalization and the operations of Narrow Syntax runs deep. Yet at no point in the normalization-driven approach has there been any need to invoke binarity of phrase structure, Copy, MOVE or movement-inducing EPP features etc. as syntactic primitives. These have been artifacts of the process of normalization as far as the previous derivations are concerned. Thus, a normalization-driven grammar may ultimately provide a formal motivation for these requirements.[27]

4.5 A WH-movement example

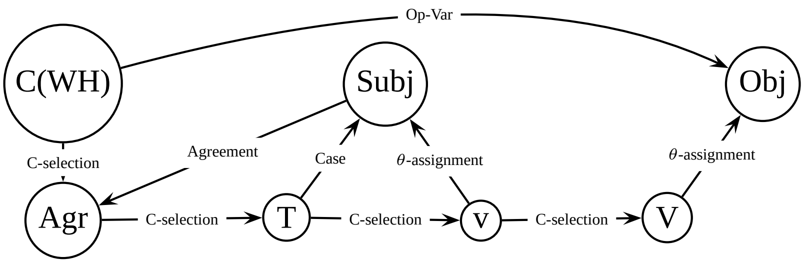
The next section illustrates how the proposed normalization-driven approach would account for an instance of WH-movement.

(50) What did Sarah eat?

This sentence includes the usual relationships (already discussed in previous sections): Selectional relationships (a–d); relationships based on feature checking by means of AGREE (e–f); argument structure relationships concerning Θ role assignment etc. (g–h); and, finally, there is the operator-variable relationship between C and the WH item (i).

(51) a. C $\rightarrow$ Agr [Selection]
b. Agr $\rightarrow$ T [“ “]
c. T $\rightarrow$ *v* [“ “]
d. *v* $\rightarrow$ V [“ “]
e. T $\rightarrow$ DP: Case licensing [AGREE]
f. DP $\rightarrow$ Agr: ϕ Agreement [“ “]
g. *v* $\rightarrow$ Subj DP: *Sarah* [Θ-roles]
h. V $\rightarrow$ Obj DP: *What* [“ “]
i. uWh (on C) $\rightarrow$ WH-item: *What* [Interrogative]

(52) a. What did Sarah eat?

b. 

The important factor which distinguishes these relations from those discussed in the previous section is the functional dependency between C and the WH-variable. The relationships in (51) and (52) can be represented by the following relation, a numeration. For the sake of convenience, we will call this Stage 1.

(53) $\dfrac{\text{Stage 1}}{(\text{C, Agr, T, } v\text{, Subj}[\phi]\text{, V, Obj})}$

This relation is in 1NF because there are no repeating attributes, all elements are parsed and all values are atomic. It is also in 2NF because all attributes are ultimately functionally dependent on C.

The relation is not in 3NF because there are a number of transitive dependencies (e.g. C $\rightarrow$ Agr $\rightarrow$ T $\rightarrow$ *v* $\rightarrow$ V $\rightarrow$ O and C $\rightarrow$ Agr $\rightarrow$ T $\rightarrow$ *v* $\rightarrow$ Subj). To solve this problem, the relation in (53) is broken up into a series of smaller relations which are in 3NF. The elements which cause the relations to fail normalization constraints are placed in new relations. For the sake of space, I will not indicate the ‘old’ relations – represented by the double strikeout in previous examples.

(54) $\dfrac{\text{Stage 2:Stage 1}}{(\text{C,Agr,}Obj_{WH})(\text{T,}v\text{,Subj})(\text{Subj}[\phi]\text{,Agr})(\text{Agr,T})(\text{T,v})(\text{v,V})\text{:(V,Obj)}}$

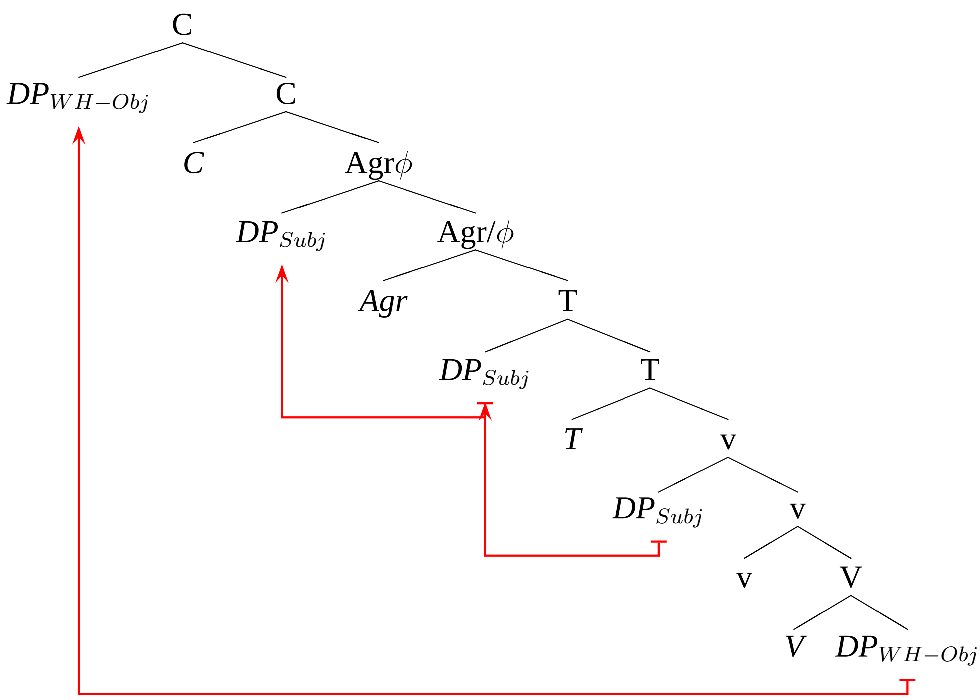
All the relations, are now in 3NF because there are no transitive dependencies in any of them.[28] All the relations, except (T *v* Subj) and (C,Agr,Obj_{WH})) are also in 4NF. However, (T, *v*, Subj) and (C,Agr,Obj_{WH}) both include multi-valued dependencies and consequently these relations are not in 4NF. To remedy this, they must be normalized; the elements that cause each relation to fail this level of normalization are removed from the original relation and placed in a new one.

(55)

Stage 3:	Stage 2:Stage 1
(C,Agr)(T,Subj)(*v*,Subj):(C,Obj_{WH})(*v*,Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(V,Obj)	

The visual representation of all these fully-normalized relations is as follows.[29]

(56)



In this way, the normalization approach derives the correct representation without having to stipulate WH-movement triggered by EPP features etc. The effect of chain formation and phrase-structure building is a function only of normalization and the relationships existing between the elements of the numeration.

4.6 *Wh*-in-situ *and covert movement*

The normalization-based approach to displacement that is sketched here requires that all non-normalized functional dependencies be normalized, most often resulting in a movement chain. This raises the question of how to deal with the overt/covert-movement distinction, for example as evidenced in multiple interrogatives where at least one WH-item remains in situ.

(57) I wonder who$_1$ bought what$_2$

In this kind of example, C has a relationship of some kind with both WH$_1$ and WH$_2$. However, whereas WH$_1$ moves overtly, WH$_2$ does not.

In approaches to questions of this type, it has been proposed that WH_2 does not actually move but is interpreted in situ by non-selective binding (Pesetsky 1987) or by a choice function (Reinhart 1998) or both (Dayal 2002). What is important is that there exist mechanisms to interpret the WH-item without it having to move. Following Reinhart (1998), I will assume that the in situ WH-item may be interpreted by a choice function.[30] Under this approach, the crucial question is whether such a choice function constitutes a functional dependency. Dayal (2002) explicitly argues that a choice function is *not* equivalent to a functional dependency. If the choice function is not a functional dependency, then it would not be necessary to normalize it (since normalization is defined exclusively in terms of functional dependencies). To illustrate how this might work, consider a simple derivation for the sentence in (57). Let us take for granted the kinds of relations listed in 51 on page 18. The relationships relevant to the actual WH question are as follows.

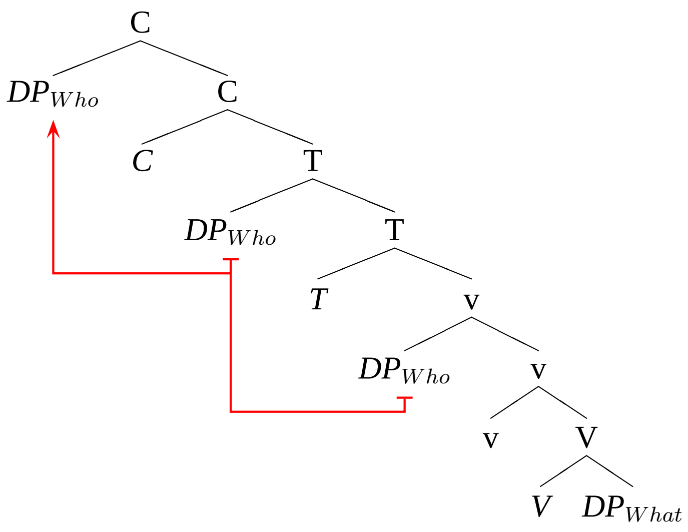
(58) I wonder who$_1$ bought what$_2$

a. uWh (on C) $\longrightarrow$ WH-item: *Who*$_1$

b. CH(f)(on C) $\rightsquigarrow$ WH-item: *What*$_2$ [Choice function]

The derivation proceeds as described in section 4.5, eventually yielding a tree like the following one where the Subject Wh-item has moved to C to satisfy the Operator on C. The set of relations is now fully normalized.

(59)



However, there is still a wh-item in situ (i.e. *what*) that needs to receive an interpretation. Since the uWH feature on C has already been checked by WH_1, there can be no feature-checking relationship between C and WH_2. Consequently, the only operation that is able to provide *What* with an interpretation is binding by means of a choice function (Reinhart 1998) in its in-situ location.[31] This paper has argued that movement only occurs in response to normalization requirements. Since the choice function is not a functional dependency (Dayal 2002), it does not affect that normalization status of the relations. Consequently, the wh-item does not need to move.[32] In this way, a functional dependency approach can capture the effects associated with multiple questions.

This approach to covert movement is not the only possible one. One option would be to follow the spirit of Kayne (1998) and claim that overt movement of WH_2 occurs but that its effects are masked by subsequent movement operations. This seems to me to be a promising avenue of research. Another option would be to reformulate cases of apparent wh-in-situ – in languages like Japanese – as involving operator movement without pied-piping the remainder of the wh-item (Watanabe 1992). Hagstrom (1998) and Bošković (2002) provide an interesting variant of this argument where a Q-morpheme moves in wh-in-situ languages. Movement of a Q-morpheme yields a pair-list reading. In the parlance of functional dependencies, there exists a dependency such that:

(60) OP (on C) → Q-morpheme → [XP]

The resulting transitive and/or multi-valued dependencies require normalization. The Q-morpheme moves, leaving the remainder of the wh-item in situ. Thus, there are a number of ways in which to interpret constituents which have not moved overtly.[33]

5 Some implications for Narrow syntax

This paper has explored the way Narrow Syntax is organized according to principles of relational databases which restrict how syntactic representations are built from the numeration. When Normal Forms are enforced for an otherwise non-binary numeration, a linguistic representation eventually emerges that is quite similar in many respects to a 'standard' tree. Thus it would appear that linguistic trees are always at least in 4NF.

Importantly, Relational Theory is mathematically well grounded and was developed outside of a linguistic and cognitive framework. Relational Theory provides well-defined and fundamental tools such as the notions of Functional Dependency and normalization. None of these are contentious within Relational Theory.

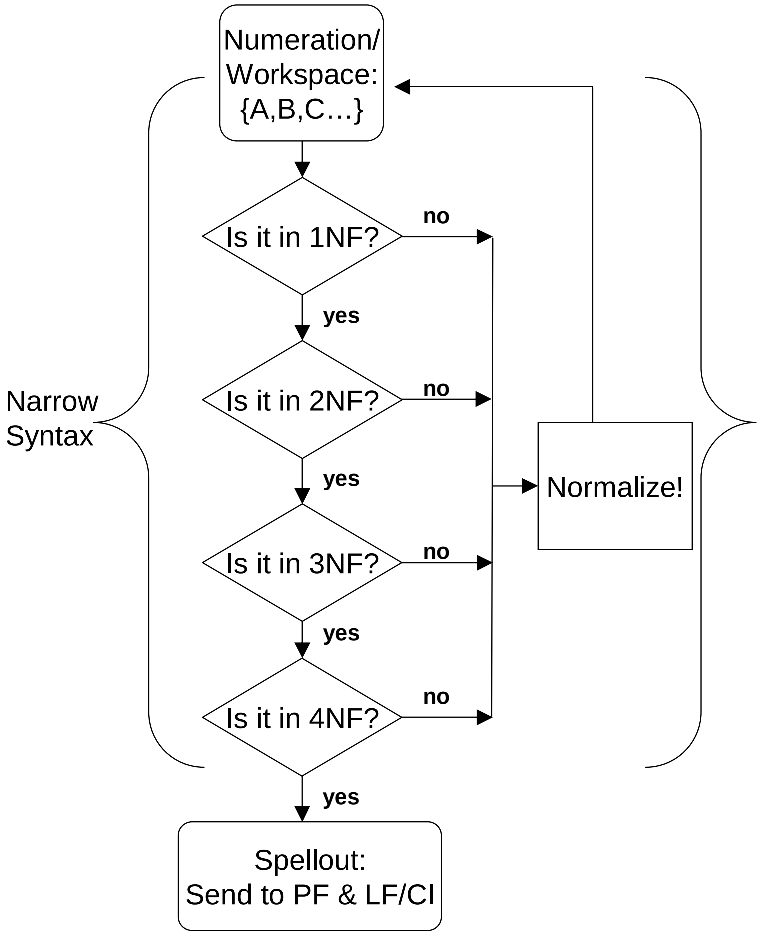
This has far-reaching consequences for the nature of syntax, some of which are outlined here. The aim of this section is not to prove or disprove these positions but merely to outline the implications of this research program and to provide an indication of the kind of data that could disprove the hypothesis (1a).[34] At the beginning of this paper, I adopted the assumption that there is a one-to-one, meaning-preserving relationship between syntactic structures and LF representations. However, by Occam's razor, there is no need for a redundant mapping so it may be possible to do away with the mapping between Narrow Syntax and LF completely. Thus, in a sense, Narrow Syntax might itself become the interface between LF and the numeration. The model is illustrated by the decision chart in figure 3 on the following page.[35]

To the extent that Narrow Syntax is driven by the interface properties of the CI interface (1a), certain Narrow Syntactic operations are derivative of extra-linguistic, but nevertheless hardwired, internal principles of normalization and may no longer be required as syntactic primitives, much less as imperfections in the system.

5.1 Displacement

With respect to MOVE, in example 49 on page 17, a single category (e.g. the XP representing the subject) is represented several times within the set of relations. In a tree diagram this is identical to the generative grammar notion of 'copy'. However, it is essential to note that at no point have

Figure 3: A decision-chart for the grammar



I introduced MOVE or chain-creation or formal features whose only reflex is to trigger movement. These effects follow simply from principles of normalization.[36] In the past, displacement has been considered an imperfection in the system (Chomsky 1995b; 1998). However, in a normalization-driven approach, displacement is an artefact of the normalization process – it is no longer an imperfection.[37][38]

5.2 Binarity

Within the Minimalist program, binarity derives from a basic MERGE operation which is argued to maximally merge two entities.[39] Binary MERGE cannot form a set with the form {a,b,c ...}. MERGE (as defined by Chomsky 1995b) is a basic syntactic primitive and is often presented as *the* primitive set-building mechanism (e.g. Chomsky (1995b)). However, there is a discrepancy within the standard Minimalist account: Minimalist approaches implicitly assume the existence of another set-building mechanism – one that is not restricted to creating binary sets – in order to create a numeration (Chomsky 1995b). The numeration is a non-binary set of elements, of the form {a,b,c...}, which must be selected from the lexicon before Narrow Syntax can commence its work. This means that in addition to (binary) MERGE, there is an even more primitive set-building mechanism, which I shall call SET MERGE, forming sets regardless of how many members there might be. This more basic mechanism is absolutely primitive and must be present in any combinatorial system.

Within the normalization-based system I have proposed, it is implicit within the definition of 1NF that SET MERGE is present in order to build a numeration. Although SET MERGE is not restricted to creating binary sets, the tree resulting from normalization in 49 on page 17 appears to be a well-formed binary branching tree. The *effect* of binarity is an artefact of normalization of non-binary sets. This means that the current approach has not had to utilize BINARY MERGE at all – which raises serious questions about its necessity.

Of course, there are contexts where normalization will not result in strictly binary branching.[40] There is a limited range of circumstances when normalization does not yield binary sets. These include instances in which a number of attributes all uniquely determine each other (i.e. circular dependencies). Although a discussion of these contexts is beyond the scope of the present paper, it is worth noting that all non-binary contexts within a normalization approach, if they are fully normalized, nevertheless conform to notions of computational non-ambiguity. This captures the original rationale for the binary stipulation in X-bar theory – namely that binarity reduces the complexity of the language learning situation. Whether the binarity stipulation of X-bar can be truly removed is a question that awaits further investigation, but the preliminary results obtained here seem to be promising.

5.3 Long-distance relationships

The normalization approach also requires all long-distance relationships which can be represented as functional dependencies to ultimately be rendered in local configurations. This is an extremely strong claim. Its implications for WH-movement have already been discussed in section (4.5). However, it also has implications (as pointed out by a reviewer) for binding: one might expect binding relationships to ultimately be reduced to local configurations.

There have been significant attempts to formulate binding in terms of movement (Cole et al.

1990, Hornstein 1999, Huang and Tang 1991, Kayne 2005, Pica 1986, Reinhart and Reuland 1991, Zwart 2002). However, there seems to be general consensus that this cannot be the whole solution. Attempting to reduce binding to movement (and thus to functional dependencies) is problematic because the domains and constraints for binding are *not* identical to those for movement. For instance, movement cannot occur out of islands, whereas binding can occur across island boundaries. In addition, the existence of subject-defined domains for Principle A, the disjunctive condition for Principle B and anti-subject-oriented pronouns in languages like Norwegian do not seem to follow in obvious ways from movement.

Binding is a complex phenomenon, possibly resulting from a set of interacting principles and/or constraints (e.g. the domain condition as well as the binding condition). While some of these conditions may be reducible to movement (or, in other words, functional dependencies), it is far from obvious that all these conditions should be reducible in this way. Thus, it seems to me premature to call for the redefinition of all binding in terms of movement.

This constitutes a challenge for the normalization approach which must be addressed within a yet-to-be-articulated broader Minimalist theory of binding. What seems clear at this stage is that, while movement may play a role (as suggested by the research quoted above), additional factors may be at play which derive the domain for binding.[41] Also, as indicated in section (4.6), there remain certain types of relationships which may fall outside the scope of functional dependency and it is possible that these could be implicated in binding.[42]

5.4 Adverbs and adjunction

For reasons of space, this paper has largely ignored the question of adjunction. Central to this issue is whether adverbs are hosted by specifiers of functional heads (Cinque 1999) or whether they are directly adjoined to phrasal projections (see for example, Ernst (2002)). The normalization approach suggests that adverbs are mediated by a cinque-style hierarchy of functional heads. Implicit in the adopted approach to phrase structure is the notion that the functionally determining element also projects.[43] Adverbs have selectional properties, and if an adverbial were to directly select, say *v*P, then that adverbial would have to project. To the extent that this is undesirable, it is necessary to have a functional head which mediates between the adverb and the projection to which it attaches.

5.5 Subject vs non-subject asymmetries

A further implication of this approach is that there is an asymmetry between subjects and non-subjects in languages with Subject-T agreement. By virtue of (a) the fact that ϕ features on the subject functionally determine $u\phi$ features on T and (b) the transitivity of functional dependency (25), the subject will functionally determine T and everything in the c-command domain of T. Non-subjects, in languages without object ϕ agreement, do not functionally determine anything. This asymmetry is predicted by any approach using functional dependencies (not only the normalization approach sketched here). It has been argued by De Vos (2006) that this asymmetry underlies phenomena such as subject-oriented anaphora in languages like Dutch, Icelandic and Chinese.

5.6 *The hierarchical nature of normalization rules*

Another prediction of the normalization approach capitalizes on the hierarchical nature of normalization rules. Normalization rules at higher levels entail normalization at lower levels. Thus, if a relation is in 3NF, it will automatically be in 2NF and 1NF. The hierarchical nature of this system seems to suggest that there will be no languages which have, for instance, multi-valued dependencies, but lack transitive dependencies.

5.7 *EPP*

Perhaps one of the most controversial implications of the normalization approach is that it challenges the existence of purely formal features such as EPP.[44] EPP features exist for the sole purpose of forcing (overt) movement to occur. In the parlance of normalization, an EPP feature instantiates a spurious functional dependency which must be normalized – thus triggering displacement. However, within a normalization approach, displacement can be the function of *any* feature (e.g. Agreement, Case etc.) and consequently, the rationale for EPP is undermined.

Furthermore, if there is a one-to-one, meaning-preserving mapping between Narrow Syntax and LF then purely formal features must have some kind of function at LF. [45]

Thus, the non-existence of EPP features (at least as they are currently conceptualized) is a (possibly refutable) prediction of the normalization approach. If it were to be proven that EPP features do exist then the normalization approach would have to be significantly revised. However, the existence of these features is currently disputed (Boeckx 2000, Martin 1999) so the jury is still out.

To a lesser extent, formal features such as Case and Agreement are also highlighted. The normalization approach would seem to suggest that these features have a LF function. For instance, Case has been shown to be related to Tense (Pesetsky and Torrego 2001) and Aspect (Svenonius 2002) and ϕ features may be related to semantic notions such as Deixis (Cowper 2005) and/or Referentiality (Reinhart and Reuland 1991) and pragmatic issues such as speaker and hearer (Harley and Ritter 2002).[46]

5.8 *Parametric variation*

The normalization proposal argues for universal constraints on LF representations and as such does not account for parametric variation. There are a number of possible avenues to account for parametric variation. First, as is standardly assumed, some parametric variation derives from the lexicon. For instance, a language lacking a particular F/uF pair will not instantiate them as a functional dependency, with consequences for the final representation. Second, as I have intimated in section (4.5) parametric variation concerned with overt/covert movement could conceivably be captured in a variety of ways by postulating (i) the existence of syntactic relationships which are technically not functional dependencies (Dayal 2002, Reinhart 1998);

(ii) movement of Q-operators without pied-piping of phonological material (Bošković 2002, Hagstrom 1998, Watanabe 1992); (iii) and the logical possibility that all movement is overt but is masked by subsequent (overt) movements (Kayne 1998).

5.9 Interim conclusions

This paper has outlined an ambitious framework embedded in Relational Theory which constrains the ways in which syntactic representations are derived from a numeration. The framework has numerous, falsifiable implications for Narrow Syntax which have been outlined in this section.

6 Two speculations

The previous sections have outlined a curious parallel between the operations of Narrow Syntax and similar processes of normalization within Relational Theory. In what follows, I wish to project forward and make certain bold speculations about the kinds of possibilities that such an approach *could* open up. To some extent, this section may be premature, but it is merely a speculation on the kinds of questions that could be posed within a normalization-driven research program.

6.1 Phases and cyclicity

This paper started with an idea about the manner in which the CI interface may constrain Narrow Syntactic operation (1a). However, (1b) has not been addressed, namely the manner in which the PF interface may affect syntactic computation. Unfortunately, this is the scope of this paper so my comments will have to be brief.

The normalization approach I have described in this paper is representational in nature and is also primarily concerned with the CI interface. As such, it does not predict the existence of phases per se – phases being at least partly concerned with the PF interface as well. However, within a normalization-driven approach, the derivation is completed in various stages and, as such, it is compatible with a phase-based approach. The following section outlines a possible way of implementing phases within a normalization-driven grammar.[47]

Within a phase-based approach to syntax (Chomsky 2001b) it is necessary that certain categories move to the 'edge' of the phase in order to be accessible to the next phase. Exactly what motivates this movement is unclear, especially if one wants to avoid looking ahead in the derivation. Usually, such 'escape-hatch' movement is formally accomplished by postulating EPP features whose sole purpose is to motivate such movement.

The normalization-driven approach promises to offer intriguing insights into escape-hatch movement to the phase Edge. There are two questions I wish to touch on: (a) what is a phase and (b) what is the 'edge'. Normalization ensures that syntactic structures are well-formed and, being free of ambiguities are interpretable at LF. Therefore, a fully-normalized structure, or sub-part of a structure, can be sent to the LF interface the moment normalization is complete (see figure 3 on page 22). The PF interface imposes stricter conditions – sets must not only be partially ordered as required by LF but, in addition, must have a total order (Kayne 1994). This leads to the definition of a phase, where I assume that Spell Out implies a simultaneous transfer to both PF and LF (Bobaljik 2002a).[48]

(61) **Normalized Phase:** A set (or syntactic constituent) is transferred to LF and PF (Spelled Out) the moment that:

a. it is at least in 4NF and

b. it has the possibility of a total order as determined by the PF interface.

The second question concerns the definition of the 'edge'. At its simplest, the Edge is that which is not able to be spelled out at a given point in a derivation. Note that this does not ascribe to the Edge any specific syntactic location (e.g. Spec *v*P or similar).

(62) **The Edge:** That part of the derivation which is not spelled out at a given application of Spell Out (although, if the derivation converges, it will ultimately be spelled out at some later application of Spell Out).

With these definitions in mind, consider the relations in example (44), reprinted here as (63).

(63)
The Edge:
(T,*v*, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(~~Agr, T, *v*, Subj,~~ V, Obj)
$\Downarrow$ **Spelled out in this cycle** $\Downarrow$
: (V, Obj)

At this point in the derivation a number of relations are totally ordered: (Subj,Agr), (Agr,T), (T,v), (v,V) and (V, Obj). However, although they are each individually ordered, they do not all exhibit a total order with respect to each other. Since (T,*v*, Subj) is yet to be normalized, it is not clear at this point how T and *v* are ordered with respect to the subject. Thus, the only relation which exhibits an unambiguous total order at this stage of the derivation is (V, Obj). It is duly spelled out, the rest of the relations remaining in the Edge. Importantly, movement to the Edge is in response *only* to the relations which an element has with other elements of the derivation.[49] The fact that elements in the Edge are not spelled out is a function of the fact that they either do not pass the normalization requirements of the CI interface (1a) or the total-order requirements of the PF interface (1b).

Normalization proceeds and the multi-valued dependencies are replaced with relations which pass 4NF. At this point, each individual relation exhibits a total ordering and consequently they are also totally ordered in relation to each other. This means that every relation can be spelled out and the Edge effectively contains nothing.

(64)
The Edge: :
:(T, Subj) (*v*, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(V, Obj)
$\Downarrow$ **Spelled out in this cycle** $\Downarrow$
:(T, Subj) (*v*, Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):

6.1.1 A WH-movement example

Now consider the example with WH-movement 54 on page 18, repeated here as 65.

(65)
The Edge:
(C,Agr,Obj_{WH})(T,*v*,Subj)(Subj[ϕ],Agr)(Agr,T)(T,v)(v,V):(V,Obj_{WH})
$\Downarrow$ **Spelled out in this cycle** $\Downarrow$
: $\emptyset$

At this point of the derivation it is important to note that the Object has moved into the Edge (along with other material) as a function of the normalization process. This is informally illustrated by the arrow. All of the relations are at least in 3NF. However, a few are also in 4NF and are candidates for Spell Out: (Subj,Agr), (Agr,T), (T,v), (v,V) and (V,Obj_{WH}). As with the previous examples, however, most of these cannot be spelled out at this point because they do not exhibit a total order with respect to each other. In addition, since (C,Agr,Obj_{WH}) is not in 4NF, there is no total order between the Object and the other constituents it is related with. Thus it is not possible to linearize the object with respect to anything. Consequently nothing can be spelled out at this point in the derivation.

During the next stage of normalization, the remaining relations are normalized to at least 4NF. Now, all the relations are fully normalized. Consequently, all the remaining sets can be spelled out at this stage. Since there is nothing which fails the normalization constraint, there is, technically speaking, nothing in the Edge.

(66)

The Edge:	:	:
:(C,Obj_{WH})(T,Subj)(*v*,Subj)	:(C,Agr) (*v*,Subj)(Subj,Agr)(Agr,T)(T,v)(v,V)	:(V,Obj_{WH})
	⇓ **Spelled out in this cycle** ⇓	
:(C,Obj_{WH})(T,Subj)(*v*,Subj)	:(C,Agr) (*v*,Subj)(Subj,Agr)(Agr,T)(T,v)(v,V)	:(V,Obj_{WH})

The result of all this is that a non-normalized relation seemingly moves to the Edge. Paradoxically, the same 'movement' to the Edge is also the same 'movement' that merges the WH-item to C. No EPP or 'Edge' features are required in intermediate positions.[50] Of course, this notion of Edges and phases is quite different to the more usual notion of the Edge as a distinct functional projection – but is nevertheless, I think, based on Minimalist principles.

This brief outline illustrates that a phase model can potentially be implemented in a normalization driven syntax although many questions await further research.

6.2 *A double dare for linguistic science*

In this paper, my primary focus has been on the relationship of Narrow Syntax to the CI interface. In the previous section, I explored the relationship to the PF interface in a phase model. In the following section I would like to adjust that focus by speculating on the relationship of Narrow Syntax to the lexicon.

One of the characteristics that separates humans from other animals is our ability to use natural language. Hauser (1996) formulates a double dare with respect to this. How do we account for the following two fundamental differences between humans and other animals:

(i) the fact that humans can create discretely infinite, recursive utterances (Chomsky 2002, Hauser et al. 2002, Hauser 1996) and

(ii) the fact that humans, with explosive speed, acquire a vast and highly organized lexicon.

Embedded recursion appears to be unique to humans (Hauser et al. 2002). For instance, humans can acquire the grammar of incrementation, allowing them to count to infinitely large numbers. Chimpanzees, in contrast, can learn a few discrete numbers and show no ability to be able to generalize the system spontaneously and only a limited ability to do so under intensive training (Hauser et al. 2000; 2002, Kawai and Matsuzawa 2000). With respect to our linguistic ability, within the Minimalist research program, recursion is ensured by a relatively impoverished system of Narrow Syntax. It has even been suggested by Hauser et al. (2002) that some of these operations could be viewed as variants of the same operation that allows incrementation to occur.

A second unique propensity of humans is the ability to rapidly acquire a huge lexicon with seemingly little effort. Even after years of intensive training, non-human animals only acquire severely restricted lexicons. Liberal estimates might allow for a few hundred tokens, with conservative estimates being considerably lower (Pinker 1994). Even granting the veracity of the larger estimates, there is still no beating humans for the sheer size of their lexicon and the speed at which it is learned. It is clearly not merely the acquisition of word lists that is important, but rather the fact that the human lexicon is uncontroversially an *organized system* of knowledge representation. Organization has direct implications for the size of the lexicon as well as the speed of its acquisition.

The size of the human lexicon and the speed at which it is acquired point towards its organization. Thus, it might be hypothesized that one of the things that sets human language apart from the symbolic systems of other communicative beasts is our distinctive cognitive ability to organize – or normalize– a lexical database. This immediately yields several advantages to the language learner. An organized (normalized) lexical database can be much larger than an unordered one, require less computational processing, can be updated or changed more easily and consequently can be acquired much faster. In my view, the seeds of a solution to Hauser's double dare lie in this idea: if it could be demonstrated that humans have the unique ability to normalize knowledge representations, then there would be an explanation for (i) the structured nature of the lexicon, (ii) its rapid acquisition (iii) its flexibility and (iv) the inability of other animals to do the same.[51]

This paper could provide evidence for this position; it has argued that the process of normalization underlies Narrow Syntax. If this is true and the human brain is capable of normalizing a subset of the lexicon (i.e. a numeration) then there is, in principle, nothing preventing normalization of the lexicon itself – an expanded numeration. This is leant credence by a wide variety of research which suggests that lexical processes mirror syntactic ones (among others Hale and Keyser 1993, Marantz 1997, Marantz and Halle 1993) and by those research traditions questioning the segregation of lexicon and syntax (Bresnan 1982, Hudson 1984, Hudson 1990, Pollard and Sag 1994). Normalization thus provides the beginning of an answer to Hauser's double challenge. Moreover, the same system that allows the rapid acquisition and organization of a complex lexicon is used to drive the syntactic component responsible for the recursive nature of language. In a sense, the lexicon comes with syntax for free.

Evidence could come from studies into comparative psychology. Normalization, at least to 3NF, requires the ability to recognize transitive dependencies and, more generally, the ability to recognize relationships between relationships. There is mixed evidence concerning transitive dependencies. Studies into animal numeracy (Hauser et al. 2000; 2002, Kawai and Matsuzawa 2000) suggest that non-human primates just do not seem to 'get it'. While they may learn numbers, there is generally little recognition that numbers are a transitive system (e.g. 8 is bigger than 6 which is bigger than 4).

Other data for transitive inference are less clear. The standard five-point test for transitive inference is as follows (Bryant and Trabasso 1971). Subjects are trained with pairs A-B, B-C, C-D and D-E. In each case, subjects are rewarded for selecting the hierarchically higher element, namely A, B, C and D for each pair. After training, subjects are presented with a novel pair, B-D. Typically, if a subject chooses B over D, then this is taken as evidence for transitive inference. A variety of non-human vertebrates have putatively demonstrated evidence of transitive inference, from squirrel monkeys (Brandan et al. 1977) and chimpanzees (Gillan 1981) to pigeons and jays (Guilermo et al. 2004, Von Fersen et al. 1991).[52] Experiments with invertebrates have been less successful (Benard and Giurfa 2004). However, a variety of papers (Allen 2006, Breslow 1981, De Lillo 1996, Russell et al. 1996, Von Fersen et al. 1991, Wasserman et al. 2001, Zentall 2001) have argued that these effects can be accounted for by cognitive processes other than inference and analogy. In short, the standard test is unreliable.

At a more general level, normalization does require the ability to distinguish relationships between relations. In a comprehensive review of the literature, Thompson and Oden (2000) argue that only apes (as opposed to monkeys) can recognize relations between relations; monkeys on the other hand, can only recognize relationships based on shared characteristics.

> There is no evidence that monkeys can perceive, let along judge relations-between-relations. This analogical conceptual capacity is found only in chimpanzees and humans. Interestingly, the “analogical ape,” like the child, can make its analogical knowledge explicit only if it is first provided with a symbol system by which propositional representations can be encoded and manipulated (Thompson and Oden 2000:363).

To illustrate, a monkey can compare a stimulus big/small triangle pair to a big/small circle pair and judge them to be ‘same’ based on their shared perceptual characteristics of relative size etc. But only an ape would be able to compare a stimulus half banana and a half apple and judge them to be ‘same’ based on having a mental representation of ‘half’.

Despite a wide consensus on this difference between monkeys and primates (Barrett et al. 2003, Gillan et al. 1981, Thompson and Oden 2000, Vonk 2003, Wright and Katz 2006), there is a debate as to whether this reflects a truly qualitative difference in animal cognitions, or whether the question is really a quantitative one. It seems to me that the jury is still out on the question of whether (a) transitive inference and, more specifically, (b) the ability to distinguish relations-between-relations is truly a feature of vertebrate cognition generally, or whether it is more characteristic of apes. Whatever the answer, there are clear differences in the ability of apes vs other vertebrates, with primates seemingly having the edge. If future research were to bear this out, then there may be evidence that the ability to compute complex analogical relations is a precondition for the ability to normalize relations, which in turn links directly to the ability to acquire a large lexicon and compute natural language.

7 Conclusions

This paper has shown that Narrow Syntax essentially reduces to the ability to normalize a numeration and has sketched the outlines for an ambitious, and yet highly minimalist, research agenda into a normalization-driven syntax. A very specific (and potentially falsifiable) view of the CI interface suggests that extra-linguistic principles of well-formedness in databases (Normal Forms) are at work in Narrow Syntax. Adopting this view yields a simplification of the model of grammar and creates an external, non-linguistic justification for its mechanics.

Principles of normalization are based on notions of computational simplicity, ease of data storage, modification and retrieval, and were originally conceived of by Codd (1970; 1983) as externalist, non-cognitive constraints. Implicit in the approach I have sketched is that these principles are strictly internalist and are thus instantiated in the mind/brain. This is thus a proposal on the specific nature of the computational system underlying language and possibly other mental modules as well.

Acknowledgements

I would like to thank two anonymous reviewers for their patience, close reading and constructive feedback on the draft versions of this article. Their input greatly improved the scope and quality of the paper. In addition, I am grateful to my supervisors at the Leiden University Centre for Linguistics and the Meertens Institute, to my friends and colleagues at Rhodes University and the LUCL for their contributions on some of the very early versions of this paper and to those who told me this idea was crazy and so – in the spirit of Niels Bohr – persuaded me it was 'crazy enough' to pursue. In addition, I must acknowledge my heartfelt thanks to Louise Todd and Layle Henderson who proofread this article. Needless to say, any remaining errors and inconsistencies are my own.

Notes

[1]In this paper, I tend to use CI/LF and SM/PF interchangeably.

[2]An anonymous reviewer has pointed out that the CI and SM interfaces can only determine Narrow Syntax if there is either a trivial *or* a one-to-one, meaning-preserving mapping from Narrow Syntax to the relevant interface. I will tentatively adopt the position that there is a one-to-one, meaning-preserving mapping. Note that if one allows for a trivial mapping then there is the possibility that at least some syntactic objects will be mapped to zero at LF.

[3]Note that it is not being claimed that all cognitive processes should be able to be modelled in terms of relational databases – merely that a part of the CI system which must interface with the language model can be modelled in this way.

[4]Crucially, the claim of this paper is that only normalized, partially ordered sets are equivalent to well-formed syntactic trees.

[5]A formal definition is as follows: "a relation R satisfies functional dependency X → Y if for every pair r_1, r_2 of tuples of R, if $r_1[X]=r_2[X]$, then $r_1[Y]=r_2[Y]$" where X and Y are attributes of R and r[X] and r[Y] are the respective values of the records. (Sagiv et al. 1981:437).

[6]Modification and deletion anomalies and project-join anomalies (Codd 1970; 1983, Dutka and Hanson 1989, Kroenke 2004).

[7]There are seven, although only the first four are considered in this paper. Also note that I treat 3NF and BCNF as a single Normal Form for the sake of simplicity. It is an extremely interesting research question whether all levels of normalization have syntactic reflexes.

[8]Alternatively, these conditions can be rephrased as operations that can be applied to any relational database in order to ensure compliance with a particular normal form.

[9]Normalization is a complex process and the toy examples utilized thus far cannot do justice to the nuances involved. Moreover, detailed discussion is beyond the scope of this paper. For further discussion, refer to the reference list.

[10]In fact, there exists 3NF which is logically, but subtly, distinct from BCNF. Due to space considerations and since BCNF is the stricter version, this is the focus of this paper. It remains a question for future research whether 3NF has linguistic reflexes.

[11]It is worth noting that there are two types of partial ordering depending on Reflexivity. If a relation is reflexive, transitive and antisymmetric then there exists a non-strict, partial ordering; if a relation is irreflexive, transitive (and

therefore antisymmetric) there exists a strict, partial ordering.

[12]The set-theoretic nature of phrase structure is also pointed out by Kayne (1994:4) who demonstrates that phrase structure does not instantiate a total ordering, and thus, by implication, must be a partial ordering.

[13]Phrase structures and functional dependencies are both transitive, antisymmetric (in terms of projection/domination) and reflexive.

[14]I take this to be equivalent to unification.

[15]If they were derivational statements, then there would probably be a violation of the extension condition.

[16]This is equivalent to the SUBCAT feature in HPSG terminology.

[17]The same general result occurs if AGREE results in feature deletion (Pesetsky and Torrego 2001). In this case, the uϕ feature will be deleted and the probe will then have the value of $\emptyset$. It is still the case that the goal determines the value (in this case $\emptyset$) of the probe. Hence, there is a functional dependency between probe and goal.

[18]Note that the Functional Dependency, illustrated informally by the dotted line, runs opposite to the direction of the probe.

[19]Throughout this paper, the terms *Subject* and *Object* are used to denote external and internal argument XPs fulfilling the derived grammatical notion of Subject and Object respectively. They are treated as being atomic attributes for the sake of simplicity, although this obviously need not be the case.

[20]I will assume that nominative case assignment to the Subject DP is part of a broader Agreement relationship between T and uT (Pesetsky and Torrego 2001).

[21]In fact, linguistic trees are also trivially in 2NF, but this will be discussed in the next section.

[22]The inverse does not necessarily hold.

[23]The structure in (38) is actually not a well-formed table – but that is because it is not in 2NF. Note that (38) would be in 2NF if XP was determined by some other element in the table.

[24]In an n-ary tree, everything c-commands everything and so all c-command and selectional relationships can be satisfied by such a tree. Note that the status of c-command is not challenged within a normalization-driven approach.

[25]Of course, there may be alternative ways of normalizing these relations although as far as I can tell they do not impact the final representation.

[26]However, within a normalization approach there is no real need to do so since the relations in (48) are perfectly well-formed as far as a CI-interface, conceived in relational terms, is concerned.

[27]An obvious problem is how to deal with the overt/covert-movement distinction and the related overt application of EPP requirements. A normalization driven approach merely derives well-formed LF structures with appropriate movement chains. Whether the head or the tail of the chain is overtly spelled out is a separate question that is presumably affected by issues such as cyclic Spell Out etc. (see suggestions by Bobaljik (2002b), Nunes (1999; 2004)). Although some speculations are in section 4.6, the normalization approach does not have insights regarding the overt application of EPP requirements except in some cases.

[28]In principle, it would also be possible to normalize these relations into the following set: (T *v* Subj), (C Obj_{WH} V), (C,Agr), (T,*v*) and (*v*, V). I do not think that anything depends on this, however.

[29]I have not included *do*-support in this tree since the normalization approach does not provide any insights at this point. Also note that the order of the sets themselves in (55) (e.g. in stage 3: (T, Subj) (C, Obj_{WH}) vs (C, Obj_{WH})(T, Subj)) is irrelevant.

[30]Dayal (2002) modulates this view by claiming that either a choice function or operator binding may be used.

[31]Presumably, languages such as Bulgarian which allow multiple feature checking would have obligatory WH movement of all wh-items to Spec CP.

[32]I am aware that Dayal (2002) argues (contra Reinhart (1998)) that Narrow Syntax can use either operator-variable binding *or* choice functions. Under my proposal here, operator-variable binding would induce movement; choice functions would not.

[33]Which of these options are attested is a subject for future research.

[34]I would like to thank an anonymous reviewer for pointing these out.

[35]An anonymous reviewer suggests that this more restrictive hypothesis raises the question of whether relations such as Case and ϕ agreement have interpretations in the CI system. Certainly, the deep question is raised as to why these kinds of features are necessary at all. This is briefly discussed in section 5.7.

[36]The proposal here is only focussed on MOVE. Other components of the 'standard' theory are not questioned here: these include Minimality, the Extension condition, the need for cyclic movement, EPP and the possibility of other interfaces, notably the PF interface, also exerting influence over Narrow Syntax.

[37]The question of Minimality remains. Short of stipulating Minimality as a basic property of grammar, the normalization-driven approach does not seem to derive Minimality from more basic considerations. A better answer must await a further articulation of phases and locality within this approach (see for example, section 6.1).

[38]A normalization-driven syntax renders the distinction between MERGE and MOVE void. This distinction is also argued against in the work of Starke (2001) who argues for a multi-domination approach with consequences for reconstruction. Both proposals also share the property that the trace of a moved category and the moved category itself must be identical. This makes it difficult to implement a solution to reconstruction along the lines of Lebeaux (1988) where adjuncts are merged relatively late in the derivation. I would like to thank an anonymous reviewer for pointing this out.

[39]But see Chametzky (2000) who questions why arguments from necessity should result in binary MERGE.

[40]It is interesting to note that Kayne (1984) disallows ternary branching only in contexts where a branch includes a governed category i.e. a small subset of transitive dependencies. In other words, the 'connectivity' approach did allow for ternary branching in some contexts – just as the normalization approach does.

[41]For a discussion of the role of functional dependencies in the definition of SUBJECT and in long-distance anaphora, see De Vos (2006).

[42]Since only functional dependencies are subject to normalization, a prediction of the normalization approach is that long-distance relationships which cannot be rendered locally are not functional dependencies.

[43]It is interesting to note that a similar, but not identical, effect is argued for in Starke (2001) as a result of diminishing the number of null functional heads. The difference between the normalization proposal and that of Starke is that for Starke, an XP in a specifier will always project its features to the mother node; in a normalization-driven syntax, a XP in a specifier will only project its features to the mother node if the XP functionally determines the adjacent node to which the mother is merged.

[44]This is only true if there is a one-to-one mapping from Narrow Syntax. If there is a trivial mapping then this does not necessarily follow since a trivial mapping can also map a feature to zero. Also note that the existence of EPP *effects* is not under dispute – merely the existence of formal EPP features.

[45]This does not prevent EPP-like effects from being artifacts of the PF-interface; the current framework merely points out that EPP cannot be a function of the LF interface.

[46]An anonymous reviewer points out that an additional question to ask is whether agreement phenomena involving these features have an interpretive dimension i.e. does the meaning of a head change when it enters into an agreement relationship? This is both an empirical and a theoretical issue. Empirically, a naïve answer might be yes e.g. the meaning of a non-finite verb is different from the meaning of a finite verb insofar as the former is not deictically anchored in terms of reference time, event time etc. Theoretically, it raises the much deeper question of why there should be such agreement relations in the first place. This is a question that I cannot answer although a normalization approach suggests that they must have some function at the CI interface.

[47]Of course, the analysis offered here can only be partial – until such time as a normalization approach can be implemented derivationally. This section merely outlines the broad brush strokes of such an implementation.

[48]I am aware that the T-model is different to the Y-model often assumed in discussions of Phases. However, nothing hinges on the distinction in the brief discussion in this paper. For instance, it is conceivable (but not necessary) that a given relation be sent to PF in phases but that the entire set of normalized relations are only sent to PF after normalization is complete.

[49]The implications of this will become clear when we examine a WH-movement example.

[50]I have placed 'movement' in quotation marks to distinguish this apparent movement to the Edge from the more usual notion of movement which creates chains of the usual kind. Also note that there may be other ways to normalize these relations which may yield slightly different results in this implementation of phases. This may be one answer to the problem of parametric variation.

[51]This position can be nuanced: for instance, it is a logical possibility that humans have the ability to implement 4NF or higher, while perhaps other species of ape could implement 2NF or perhaps even 3NF. This is an open research question – and an exciting one.

[52]This contrasts with Piaget who claimed that transitive inference is a property of the operational stage and only develops in children as late as 8 to 10 years old (Breslow 1981, Bryant and Trabasso 1971).

References

Allen, Collin. "Transitive Inference in Animals: Reasoning or Conditioned Associations?" In *Rational Animals?*, edited by S. Hurley, and M. Nudds, Oxford: Oxford University Press, 2006, 175–185.

Armstrong, W. "Dependency Structures of Database Relationships." *Proceedings of IFIP* 74: (1974) 580–583.

Barrett, Louise, Peter Henzi, and Robin Dunbar. "Primate Cognition: From 'what Now?' to 'what If?'." *Trends in Cognitive sciences* 7: (2003) 494–497.

Benard, Julie, and Martin Giurfa. "A Test of Transitive Inferences in Free-Flying Honeybees: Unsuccessful Performance Due to Memory Constraints." *Learning and Memory* 11: (2004) 328–336.

Bobaljik, Johathan. "Realizing Germanic Inflection: Why Morphology Does Not Drive Syntax." *Journal of comparative Germanic linguistics* 6: (2002a) 129–167.

Bobaljik, Jonathan. "A-Chains at the PF-Interface: Copies and ''covert" Movement." *Natural Language and Linguistic Theory* 20: (2002b) 197–267.

Boeckx, Cedric. "EPP Elliminated.", 2000. M.s. University of Connecticut, Storrs (http://www.sinc.sunysb.edu/Clubs/nels/jbailyn/eppeliminated.pdf (.))

Bošković, Željko. "On Multiple WH-fronting." *Linguistic Inquiry* 33: (2002) 351–383.

Brandan, Xxx., O. McGonigle, and M. Chalmers. "Are Monkeys Logical?" *Nature* 267: (1977) 694–696.

Breslow, Leonard. "Reevaluation of the Literature on the Development of Transitive Inferences." *Psychogical buletin* 89: (1981) 325–351.

Bresnan, Joan. *The Mental Representation of Grammatical Relations*. Cambridge MA: MIT Press, 1982.

Bryant, P., and T. Trabasso. "Transitive Inference and Memory in Young Children." *Nature* 232: (1971) 456–458.

Chametzky, Robert. *Phrase Structure : From GB to Minimalism*. Malden MA: Blackwell, 2000.

Chomsky, Noam. "A Minimalist Program for Linguistic Theory." In *The View from Building 20: Essays in Linguistics in Honour of Sylvain Bromberger*, edited by Kenneth Hale, and Samuel Jay Keyser, Cambridge MA: MIT Press, 1993, 1–52.

———. "Bare Phrase Structure." In *Government and Binding theory and the Minimalist Program*, edited by Gert Webelhuth, Cambridge MA: Basil Blackwell, 1995a, 383–439.

———. *The Minimalist Program*. Cambridge MA: MIT Press, 1995b.

———. *Minimalist Inquiries: The Framework*. MIT Occasional Papers in Linguistics, 15, 1998.

———. "Minimalist Inquiries: The Framework." In *Step by Step: essays on minimalist syntax in honor of Howard Lasnik*, edited by Roger Martin, David Michaels, and Juan Uriagereka, Cambridge MA.: MIT Press, 2000, 89–155. Reprinted.

———. "Beyond Explanatory Adequacy." In *MIT Occasional Papers in Linguistics,* Cambridge MA: MIT Press, 2001a, number 20 in MIT Occasional Papers in Linguistics.

———. "Derivation by Phase." In *Ken Hale: A Life in Language*, edited by Michael Kenstowicz, Cambridge MA.: MIT Press, 2001b, 1–52.

———. "Language and the Brain." In *On Nature and Language*, edited by Adriana Belletti, and Luigi Rizzi, Cambridge, UK.: Cambridge University Press, 2002, 61–91.

Cinque, G. *Adverbs and Functional Heads: A Cross-Linguistic Perspective*. New York: Oxford

University Press, 1999.

Codd, E. F. "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM* 13, 6: (1970) 377–387. URL (http://dblp.uni-trier.de/db/journals/cacm/Codd70.html (.)) Also published in/as: 'Readings in Database Systems', M. Stonebraker, Morgan-Kaufmann, 1988, pp. 5–15.

KEY: codd70
ANNOTATION: Shows problems of hierarchical data organization, presents a non-hierarchical 'normal' structure, and discusses languages for describing relations of such normal structures.

———. "A Relational Model of Data for Large Shared Data Banks (Reprint)." *Communications of the ACM* 26, 1: (1983) 64–69. Reprint of Codd (1970).

Cole, Peter, Gabriella Hermon, and Li-May Sung. "Principles and Parameters of Long-Distance Reflexives." *Linguistic Inquiry* 21: (1990) 1–22.

Cowper, Elizabeth. "The Geometry of Interpretable Features: INFL in English and Spanish." *Language* 81: (2005) 10–46.

Dayal, Veneeta. "Single-Pair versus Multiple-Pair Answers: Wh-in-Situ and Scope." *Linguistic Inquiry* 33: (2002) 512–520.

De Lillo, Carlo. "The Serial Organization of Behaviour by Non-Human Primates; an Evaluation of Experimental Paradigms." *Behavioural Brain Research* 81: (1996) 1–17.

De Vos, Mark. "SUBJECTHOOD and Phrase Structure: Evidence from Long-Distance Anaphora." In *Proceedings of SICOL 2006*. Seoul: Linguistic Society of Korea, 2006, 219–228.

Devlin, Keith. *The Joy of Sets: Fundamentals of Contemporary Set Theory*. New York: Springer, 1993.

Dutka, Alan, and Howard Hanson. *Fundamentals of Data Normalization*. Reading MA: Addison-Wesley Publishing Company, 1989.

Epstein, Samuel. "Unprincipled Syntax and the Derivation of Syntactic Relations." In *Working Minimalism*, edited by Samuel Epstein, and Norbert Hornstein, Cambridge MA: MIT Press, 1999, 317–345.

Ernst, T. *The Syntax of Adjuncts*. Number 96 in Cambridge Studies in Linguistics. Cambridge, UK: Cambridge University Press, 2002.

Gillan, Douglas. "Reasoning in the Chimpanzee: II. Transitive Inference." *Journal of experimental psychology: Animal behavior processes* 7: (1981) 150–164.

Gillan, Douglas, David Premack, and Guy Woodruff. "Reasoning in the Chimpanzee: I. Analogical Reasoning." *Journal of experimental psychology: animal behavior processes* 7: (1981) 1–17.

Guilermo, Paz-Y-Miño, Alan Bond, Alan Kami, and Russell Balda. "Pinyon Jays Use Transitive Inference to Predict Social Dominance." *Nature* 430: (2004) 778–781.

Hagstrom, Paul. *Decomposing Questions*. Ph.D. thesis, MIT, 1998.

Hale, Kenneth, and Samuel Jay Keyser. "On Argument Structure and the Lexical Expression of Syntactic Relations." In *The View from Building 20*, edited by Kenneth Hale, and Samuel Jay Keyser, Cambridge MA: MIT Press, 1993, 53–109.

Halmos, Richard. *Naive Set Theory*. Princeton: Van Norstrand, 1960.

Harley, H., and E. Ritter. "Person and Number in Pronouns: A Feature-Geometric Analysis." *Language* 78, 3: (2002) 482–526.

Hauser, Marc, Susan Carey, and Lilian Hauser. "Spontaneous Number Representation in Semi-

Free-Ranging Rhesus Monkeys." *Proceedings of the Royal Society-London: Biological sciences* 829–833.

Hauser, Marc D., Noam Chomsky, and Tecumseh Fitch. "The Faculty of Language: What is It, Who Has It and How Did It Evolve?" *Science* 298: (2002) 1569–1579.

Hauser, Mark. *The Evolution of Communication*. Cambridge MA: MIT Press, 1996.

Hornstein, Norbert. "Movement and Control." *Linguistic Inquiry* 30: (1999) 69–96.

Huang, C.-T. James, and C.-C. Jane Tang. "The Local Nature of the Long-Distance Reflexive in Chinese." In *Long-Distance Anaphora*, edited by Jan Koster, and Eric Reuland, Cambridge: Cambridge University Press, 1991, 263–282.

Hudson, Richard. *English Word Grammar*. Oxford: Blackwell, 1990.

Hudson, Richard A. *Word Grammar*. Oxford, UK: B. Blackwell, 1984. URL (citeseer.ist.psu.edu/hudson98word.html (.))

Kawai, N., and T. Matsuzawa. "Numerical Memory Span in a Chimpanzee." *Nature* 403: (2000) 39–40.

Kayne, Richard. "Unambiguous Paths." In *Connectedness and Binary Branching*, edited by Richard Kayne, Dordrecht: Foris, 1984, 129–163.

———. *The Antisymmetry of Syntax*. Cambridge MA: MIT Press, 1994.

———. "Overt vs. Covert Movement." *Syntax* 1, 2: (1998) 128–191.

———. *Movement and Silence*. Oxford: Oxford University Press, 2005.

Kroenke, D. *Database Processing: Fundamentals, Design & Implementation*. New Jersey: Prentice Hall, 2004, 9 edition.

Lebeaux, D. *Language Acquisition and the Form of the Grammar*. Ph.D. thesis, University of Massachusetts, Amherst MA, 1988.

Marantz, Alec. "No Escape from Syntax: Don't Try Morphological Analysis in The Privacy of Your Own Lexicon." In *University of Pennsylvania Working Papers in Linguistics*, edited by Alexis Dimitriadis, Laura Siege, Clarissa Surek-Clark, and Alexander Williams, Philadelphia: Penn Linguistics Club, 1997, 201–225. Proceedings of the 21st Annual Penn Linguistics Colloquium.

Marantz, Alec, and Morris Halle. "Derivational Morphology and the Pieces of Inflection." In *The View from Building 20: Essays in Linguistics in Honour of Sylvain Bromberger*, edited by Kenneth Hale, and Samuel Jay Keyser, Cambridge MA: MIT Press, 1993, 111–176.

Martin, Roger. "Case, the Extended Projection Principle and Minimalism." In *Working Minimalism*, edited by Samuel Epstein, and Norbert Hornstein, Cambridge MA: MIT Press, 1999, 1–25.

Nunes, Jairo. "Linearization of Chains and Phonetic Realization of Chain Links." In *Working Minimalism*, edited by Samuel Epstein, and Norbert Hornstein, Cambridge MA: MIT Press, 1999, 217–250.

———. *Linearization of Chains and Sideward Movement*. Cambridge MA: MIT Press, 2004.

Pesetsky, D. "Wh-in-Situ: Movement and Unselective Binding." In *The Representation of (in)Definiteness*, edited by E. Reuland, and A. Ter Meulen, Cambridge MA: MIT Press, 1987, 98–129.

Pesetsky, David, and Esther Torrego. "T-to-C Movement: Causes and Consequences." In *Ken Hale: A Life in Language*, edited by Michael Kenstowicz, Cambridge MA: MIT Press, 2001, 355–426.

Pica, P. "On the Nature of the Reflexivization Cycle." In *Proceedings of NELS 17, GLSA*, edited by J. McDonough, and B. Plunkett. Amherst MA: University of Massachusetts, 1986, 483–499.

Pinker, Steven. *The Language Instinct : The New Science of Language and Mind*. London: Penguin, 1994.

Pollard, Carl, and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. Chicago, Illinois: University of Chicago Press and CSLI Publications, 1994. URL (citeseer.ist.psu.edu/pollard94headdriven.html (.))

Reinhart, Tanya. "Wh-in-Situ in the Framework of the Minimalist Program." *Natural language semantics* 6: (1998) 29–56.

Reinhart, Tanya, and Eric Reuland. "Anaphors and Logophors: An Argument Structure Perspective." In *Long-Distance Anaphora*, edited by Jan Koster, and Eric Reuland, Cambridge: Cambridge University Press, 1991, 283–322.

Russell, James, Teresa McCormack, Julie Robinson, and Grainne Lillis. "Logical (versus Associative) Performance on Transitive Reasoning Tasks by Children: Implications for the Status of Animals' Performance." *The quarterly journal of experimental psychology* 49B: (1996) 231–244.

Sagiv, Yehoshua, Claude Delobel, D. Stott Parker, and Ronald Fagin. "An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic." *Journal of the Association for Computing Machinery* 28: (1981) 435–453.

Starke, M. *Move Disolves Into Merge: A Theory of Locality*. Ph.D. thesis, University of Geneva, 2001.

Svenonius, Peter. *Subjects, Expletives and the EPP*. Oxford: Oxford University Press, 2002.

Thompson, Roger, and David Oden. "Categorical Perception and Conceptual Judgments by Nonhuman Primates: The Paleological Monkey and the Analogical Ape." *Cognitive Science: A Multidisciplinary Journal* 24: (2000) 363–396.

Uriagereka, Juan. "Multiple Spell-Out." In *Working Minimalism*, edited by Samuel Epstein, and Norbert Hornstein, Cambridge MA: MIT Press, 1999, 251–282.

Von Fersen, Lorenzo, C. Wynne, and Juan. Delius. "Transitive Inference Formation in Pigeons." *Journal of experimental psychology: animal behavior processes* 17: (1991) 334–341.

Vonk, Jennifer. "Gorilla (Gorilla Gorilla Gorilla) and Orangutan (Pongo Abelii) Understanding of First- and Second-Order Relations." *Animal Cognition* 6: (2003) 77–86.

Wasserman, Edward, Joël Fagot, and Michael Young. "Same-Different Conceptualization by Baboons (Papio Papio): The Role of Entropy." *Journal of comparative psychology* 115: (2001) 42–52.

Watanabe, Akira. "Subjacency and S-structure Movement of Wh-in-Situ." *Journal of East Asian Linguistics* 1: (1992) 255–291.

Wright, Anthony, and Jeffrey Katz. "Mechanisms of Same/Different Concept Learning in Primates and Avians." *Behavioural Processes* 72: (2006) 234–254.

Zentall, Robert. "The Case for a Cognitive Approach to Anima Learning and Behaviour." *Behavioural processes* 54: (2001) 65–78.

Zwart, Jan-Wouter. "Issues Relating to a Derivational Theory of Binding." In *Derivation and Explanation in the Minimalist Program*, edited by Samuel Epstein, and T. Seely, Malden MA: Blackwell, 2002, 269–304.