Chapter 1
Introduction: Structure and its Interpretation

“Suppose I wanted to have a party?” I said.
“Like, what kind of a party?”
“Suppose I wanted Noam Chomsky explained to me by two girls?”
“Oh, wow.”
“If you’d rather forget it...”
“You’d have to speak with Flossie,” she said. “It’ll cost you.” Now was the time to tighten the screws. I flashed my private investigator’s badge and informed her it was a bust.

—Woody Allen, “The Whore of Mensa”

1.1 Begin the Begin
This thesis proposes a theory, taking advantage of mechanisms of the Minimalist Program, to replace Control Theory. This proposal uses independently motivated requirements of the Minimalist Program to derive the properties of a large class of null subjects in non-finite clauses—more often explained by the hypothesis of the phonologically-null anaphoric pronominal PRO, whose interpretation is given by Control Theory. Insofar as this proposal is simpler and contains fewer stipulations than Control Theory, as well as having adequate empirical coverage, PRO and Control Theory become redundant and can be eliminated.

This first chapter is an introduction to the rest of the thesis, and summarizes the syntactic research relevant to it. This chapter is organized as follows. There is a general overview of the Minimalist Program (MP) in §1.2 and of previous versions of Control Theory in §1.3. In §1.4 we lay the groundwork for the remainder of this thesis by describing the motivations and the basic structure of the arguments to follow. Finally there is a roadmap of the rest of the thesis in §1.5.
1.2 The Minimalist Program

The Minimalist Program grew out of the efforts of researchers in the Principles and Parameters framework. Earlier theories in the Principles and Parameters framework, such as Government and Binding Theory as presented in Chomsky (1981, 1982, 1986a, 1986b), include a rich set of principles from which it is possible to deduce logically the grammaticality of an utterance. In the Minimalist Program, as presented in Chomsky (1993), it was attempted to simplify the theory of the syntax of natural language to the greatest possible extent. However, at this time the Minimalist Program is still highly conceptual, and there are a wide range of proposals within the Minimalist Program which address syntactic problems in strikingly different ways. For that reason, the version of the Minimalist Program which I will outline is only the version with which I feel most comfortable. The main differences between Chomsky (1993) and the version of the Minimalist Program presented here are presented and analyzed in detail in Ferguson and Groat (1994, 1995) and Ferguson (1996).

The main components of the grammar include the structure-building operations of Merge and Move, feature checking through movement to functional categories, the interfaces and economies of derivation and representation

1.2.1 Structure-building via Generalized Transformations.

Chomsky (1993) assumes that syntactic structures are built iteratively, by taking smaller structures and combining them by a computational procedure, starting with a multiset of items drawn from the lexicon. This computational procedure takes lexical items — atomic syntactic structures — in addition to more complex structures and combines them into larger structures.

The “base case”—the bottom level of structure building—are the lexical or functional heads. From them, larger syntactic structures are built. In particular, given phrase structures
P and Q, we can combine them into a larger structure R with the structure \([ R \ P Q]\), with labels given to R, P and Q in a way consistent with X´-Theory. There are two types of structure-building operations, called in Chomsky (1993) *Merge* and *Move*. Schematically, the Merge operation may be illustrated as in (1):

(1) **Merge:**

\[
\begin{array}{c}
\triangle P \\
\triangle + \triangle Q \\
\rightarrow \triangle R
\end{array}
\]

P and Q may be lexical items or complex phrases, and Merge combines them to create a new structure R. The operation Move takes a substructure Q of P and creates a new phrase R which has P and a copy of Q as immediate substructures, again requiring the labels P, Q and R be consistent with X´-Theory, as seen below.

(2) **Move:**

\[
\begin{array}{c}
\triangle P \\
\triangledown Q \\
\rightarrow \triangle R
\end{array}
\]

The sole difference between the Merge and Move operations is that the definition of Merge requires the P and Q be separate structures — that P does not contain Q and Q does not contain P — while the definition of Move requires either that P contains Q or that Q contains P. From this, it is clear that the definitions of Merge and Move are complementary, and therefore both Merge and Move are the artificially-divided halves of a single structure-building operation.

Also, notice that the operation of Move as schematized above leaves a trace of Q
where \( Q \) was before the Move operation.\(^1\) Between \( Q \) and \( t_0 \) a chain \( \{Q, t_0\} \) may be formed. This chain subsumes the two categories: \( Q \) and \( t_0 \) are no longer independent objects of the syntax, but only exist as part a single syntactic object—the chain \( \{Q, t_0\} \).

### 1.2.2 \( X' \)-Theory and Structural Domains

Most of the modules of the theory are interface conditions, which apply only to the output of the syntax at PF or LF. These will be discussed in §1.2.4. However, there are a few modules which hold of each operation of structure-building. One of them is \( X' \)-Theory, which constrains what kinds of categories can be projected by heads. \( X' \)-Theory, originally proposed by Jackendoff (1977), has been subject to radical reconsideration recently, for example in Kayne (1994) and Chomsky (1994), who attempt to deduce \( X' \)-Theory from more basic principles. However, I will still use the older and possibly redundant earlier theory. There are a number of ways to formalize \( X' \)-Theory, but I will choose the following, because it is typical of \( X' \) proposals and quite straightforward.

In the standard form of \( X' \)-Theory, there are three varieties of phrases: the head \( X' \), the intermediate category \( X' \) which immediately dominates the head, and the phrase \( XP \) which immediately dominates \( X' \). The head is said to *project* the other phrasal types, and the head’s features are shared by those other categories. Any category and its immediate daughters must be of the form given in (3):

(3) a. \[ \begin{array}{c}
X' \\
\downarrow \\
X' \quad YP
\end{array} \]

b. \[ \begin{array}{c}
XP \\
\downarrow \\
YP \quad X'
\end{array} \]

\(^1\)Whether the trace is a copy of \( Q \) or something else, it is not necessary to determine here, as long as a chain is properly formed. How the trace is pronounced is determined by the PF interface, just as the semantic interpretation of the trace is determined at LF.
The Merge and Move operations described above in (1-2) must create structures satisfying (3a-d). As a result, every operation of Merge or Move must match one of the structural elements in (3). For historical reasons, operations which are like (3a-b) are called substitution operations, and operations which are like (3c-d) are called adjunction operations. As we will see below, there are basic differences between substitution and adjunction operations with respect to economy metrics.

The sister node of X˚ in (3a) is commonly called the complement of X˚, and the sister of X˚ in (3b) the specifier. However, we can sharpen these concepts by defining the Minimal Domain. Consider the following local tree in (4), from Chomsky (1993). It is important to note that the category H, adjoined to the head X˚ could only have arrived there by a Move operation; therefore H heads a chain {H, ..., t}, and only this chain can enter into the structural relationships which will be discussed here.
For a head $\alpha$, take $\text{MAX}(\alpha)$ to be the least full-category maximal projection containing $\alpha$. In (4), $\text{MAX}(H) = \text{MAX}(X) = \{XP_1, XP_2\}$, both segments of the category $XP$. The domain of a head $\alpha$ is the set of nodes contained in $\text{MAX}(\alpha)$ which are distinct from and do not dominate $\alpha$. In this example, the domain of $X$ is $\{UP, ZP, WP, YP, H\}$ and whatever they dominate. We can divide the domain of $\alpha$ into the complement domain and the residue of the complement domain, which includes the checking domain. The complement domain of $X$ in (4) is the subset of the domain dominated by the complement of the head (including the complement itself), which is just $YP$ and whatever it dominates. The residue is just the remainder of the domain: $\{UP, ZP, WP, H\}$ and whatever they dominate. For all of these definitions, though, we are interested in the minimal sets; that is, we only care about the domain categories which are closest to the head, in the sense that, for those categories, there is no category which dominates them which is also in the domain. Therefore, the Minimal Domain of $X$ in (4) is $\{UP, ZP, WP, YP, H\}$, the Minimal Complement Domain is $\{YP\}$, and the Minimal Residue is $\{UP, ZP, WP, H\}$.

These concepts can be naturally extended to include (multiple-member) head chains as well as single head positions. For instance, the Domain for a head chain $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is everything dominated by the least full-category maximal projection which dominates every member of the chain, not including the chain members and everything which dominates the chain members. The Minimal Domain is therefore the subset of the Domain which is not contained within another element of the Domain, and the definitions of the other structural concepts follows trivially.

1.2.3 Functional Categories, Feature-checking and Shortest Move

The core operation of the Minimalist Program is feature-checking through movement to a checking domain. Certain kinds of features must be checked, that is, licensed by being in a certain structural configuration with a head which is lexically allowed to check that
feature by virtue of its features. In general this is done by moving the category marked
for that feature to the specifier of a head which can check that feature, that is, into the
head’s checking domain. Thus, the need for features to be checked can motivate movement,
which creates chains.

In this work we will be concerned with the chains brought about by the checking of
several features, primarily the features of Case, Agreement, and Wh. Phrases which bear
these features are required to move to the specifier of a head which can check these
features: Case is checked by V° or T°, Agreement by Agr°, and Wh by C°. (This of course
assumes the “split-Infl” system of Pollock 1989.)

Part of the definition of movement is the “Shortest Move” requirement. This condition
only allows feature-checking movement operations which are as local as possible. The
basic set of definitions which enable this requirement are given as follows:

(5) **Minimal Link Condition** (Chomsky 1995: p.296, ex.82)

   \[ \alpha \text{ can raise to target } K \text{ only if there is no legitimate operation } \text{ Move } \beta, \text{ where } \beta \text{ is}
   \text{ closer to } K. \]

(6) **Equidistance** (Chomsky 1995: p. 184, ex. 15)

   If \( \alpha \) and \( \beta \) are in the same minimal domain, they are equidistant from \( \gamma \).

It is critical to note with respect to (6) that Equidistance can apply to the “take-off” or the
“landing” site.² If \( \alpha \) or \( \beta \) are equidistant, then either \( \alpha \) or \( \beta \) can move to a c-commanding
category \( \gamma \), or either \( \alpha \) or \( \beta \) can receive a movement by a category \( \gamma \) which both of them

²This is made explicit in other definitions of Equidistance in Chomsky (1995). E.g. p. 356, ex. 190:
   (i) \( \beta \) is closer to \( K \) than \( \alpha \), \( \beta \) c-commanding \( \alpha \), unless \( \beta \) is in the same minimal domain as \( \alpha \)
   target of movement, or (b) \( \alpha \).
   In this definition, (ia) refers to equidistant “landing” positions, and (ib) to equidistant “take-off” positions.
There are other approaches to the Shortest Move Requirement. In particular, Ferguson and Groat (1995) and Ferguson (1996) dispense with Equidistance, and instead define “closer” with respect to checkable features, i.e. a “feature-relativized shortest move requirement.” In this analysis, it is the features of c-commanding heads which determine the Minimality domains for movement. Therefore, for Ferguson and Groat (1995) there is no deep difference between object Case-checking movement to SpecAgrOP and object noun incorporation. It is only a language-particular choice what sort of feature checking to allow: the type of movement (head-adjunction versus specifier movement), and the relevant feature-checking head (bare V° versus the composite head AgrO° + V°). For working out the details of various types of feature-checking movement, including object movement to SpecAgrOP, as well as the superiority of the SMR to the notion of “Equidistance” in Chomsky (1993), see Ferguson and Groat (1995). However, in this thesis we will not consider this variant of the Shortest Move Requirement.

Finally, for expository clarity the English functional projections Tense and AgrS are fused into Infl, following Thráinsson (1996). This may be done without any loss of generality, since in English T always moves to AgrS, and SpecTP is not available as a landing site (for this, and for other languages where SpecTP is a possible landing site, see Bobaljik and Jonas 1996).

1.2.4 Legitimate Structural Descriptions, Convergence and Economy.

In the Minimalist Program, two “interface” levels are recognized: the articulatory-phonetic interface PF, and the conceptual-interpretative interface LF. A grammatical utterance is associated with the pair \{PF\_1, LF\_1\}, where the pair is licensed according to the grammar.

Structure-building by the Generalized Transformations of Merge and Move proceeds, obeying the strict cycle, until Spell-Out, where the derivations of PF and LF diverge. Certain syntactic features must be checked before Spell-Out for phonological well-formedness; these features must be checked “overtly”. However, the principle of
Procrastinate requires that features which do not need to be checked overtly must be checked “covertly”—i.e. after Spell-Out and before LF. At LF all relevant features must already have been checked. A derivation yields a legitimate structural description at LF only if that structural description is made entirely of “legitimate” objects. Only legitimate objects have an interpretation at the (PF or LF) interface, and so this requires that the LF interface be made entirely of interpretable objects. Among other conditions, a legitimate objects’ features are required to have been checked, since unchecked syntactic features are not interpretable at LF. If so, the derivation is said to “converge”, and if not, the derivation is said to “crash” at the interface. Also, the “Greed” economy condition requires all movement operations to be required to check a feature.

Besides Procrastinate, there are other global Economy considerations which take a set of convergent LF derivations within a particular equivalence class and compare them; only the most “economical” is an acceptable derivation—in terms of Greed and length of derivation, say. It is unclear whether global economy constraints are the best way to approach these issues. Since these questions will have little practical importance in this thesis, they will not be pursued here.

1.2.5 Conclusion.

Perhaps more important than describing what it part of the Minimalist Program is noticing what is not a part of it. First, although languages may differ on whether a particular operation (say, object Case checking) is overt or covert, the Minimalist Program assumes that these operations are obligatory in all languages. Any truly optional operation (e.g. topicalization) poses a problem for the Minimalist Program, although in some cases it can

---

3This is the principle of “Full Interpretation” under a slightly different guise. However, a subtle point arises here: is it possible for a object with a null interpretation (an expletive, say) to exist at LF? Chomsky (1993) claims not—expletives must be deleted at the LF interface, because they have no interpretation. However, there is a difference between not having an interpretation and having a null interpretation. Whether this is a difference which makes no difference (to quote the estimable Scotty) is yet to be seen.

be dealt with by “pushing” the optionality from the transformational component into the lexical component. Moreover, we have not discussed several topics which usually form the bulk of the exposition in a Principles and Parameters Theory: Theta Theory, Binding Theory, and Control Theory. All three of these concern the interpretation of syntactic structures, and the sort of mechanisms required for them are outside the core operations of the Minimalist Program (not that those operations could not be liberalized). However, in the Minimalist Program it is usually assumed that all of these theories hold at LF, by default.

Also, there have been a number of recent additions to the Minimalist Program which I will not be adopting in this thesis. As mentioned above, I will assume a version of X’-Theory, instead of replacing it with either Kayne’s (1994) LCA or with the Bare Phrase Structure of Chomsky (1994). This is done only for expository convenience. Also, the suggestions of Chomsky (1995) in “Categories and Transformations” concerning functional categories, especially Agr, will not be adopted here. I believe it to be wrong, because it assumes that Agr has no interpretation, an assumption which is incorrect according to Adger (1994) and references therein. Therefore I will maintain the typology of functional categories of Chomsky (1993), including Agrs and AgrO.

This description of the Minimalist Program is necessarily sketchy, but will allow us to discuss the parts which most closely relate to this thesis: how syntactic derivation proceeds to LF, and the kinds of structures one finds at LF.

1.3 Previous Approaches to Control Theory

The proper formulation of Control Theory has always posed a problem in the theory of grammar. Structural theories (Chomsky 1986a), thematic theories (Nishigauchi 1984), Binding Theory analyses (Bouchard 1983, Manzini 1983), semantic theories (Jacobson 1992) and pragmatic theories have all been proposed to explain the properties of the interpretation of PRO.
Although theories of control primarily address the distribution of PRO (that is, where structurally PRO may appear), in a complete theory of grammar it is also necessary to discuss the interpretation of PRO (i.e. what other object in the sentence PRO is coindexed with). Both questions have proven to be difficult, but there are two approaches which have attracted the most attention: the Binding theoretic approach of e.g. Chomsky (1981, 1982), and the Case theoretical approach of e.g. Bošković (1995).

1.3.1 Binding Theoretic Approaches: Chomsky (1981)

In Government and Binding Theory, the Theta Criterion and the Projection Principle together require that infinitival clauses have a subject, even though that subject is not phonologically overt. GB Theory assumes that PRO is the subject in those clauses, but this begs the question of why PRO cannot be the subject of other types of clauses. In other words, why are sentences like the following not observed?

(7)  a. *PRO sees Jamie
     b. *John believes PRO to see Jamie

In (7a), PRO is the subject of a finite clause, and in (7b) PRO is the subject of a raising clause, both of which are starred. However, to explain these cases and others, GB Theory had a appealing explanation.

First, consider the typology of phonologically null categories in GB Theory with respect to the Binding Theory of Chomsky (1981). Traces of A-movement are subject to Condition A, pro (“little pro”) is subject to Condition B, and traces of A´-movement are subject to neither Condition A or Condition B. Then, there remains only one type of null category — PRO — and only one set of Binding Theory Conditions — subject to both Condition A and Condition B. Now, consider the Binding Theory of Chomsky (1981):
(8) a.  *Condition A*
Anaphors must be bound within their governing category.

b.  *Condition B*
Pronouns must be unbound within their governing category

c.  *Condition C*
R-expressions must be free.

(9)  *Governing Category*
The governing category of \( \alpha \) is the least category which dominates \( \alpha \), a governor of \( \alpha \), and a SUBJECT accessible to \( \alpha \).

At first glance, Condition A and Condition B are flatly contradictory, and therefore the question is posed how PRO could ever be present if it must satisfy both of them. However, Chomsky (1981) noticed that PRO could indeed satisfy both Condition A and Condition B if and only if PRO did not have a governing category, which ultimately requires that PRO not have a governor. This result is referred to as the “PRO theorem.” Under the assumptions of GB Theory, in which \( C^\circ \) is a “defective governor”, PRO is therefore only able to occur in non-finite clauses embedded under a CP, as in (9).

(10)  \([_{CP} \ C^\circ \ [_{IP} \ PRO \ I^\prime\{to\} \ VP]]\]

It is important to note that this means that PRO cannot occur in “bare” IP’s in the configuration \([_{V} \ V^\circ \ [_{IP} \ PRO \ to \ VP]]\), because here PRO would be governed by the matrix verb. This is necessary because this configuration is precisely that of exceptional Case marking (ECM) in GB Theory.

Although GB Theory could in this way explain the distribution of PRO, there was no correspondingly generally accepted theory of the interpretation of PRO. In other words, there was not a complete “Control Theory”. Also, the PRO theorem depends critically on
the details of the Binding Theory. If, for instance, Conditions A and B do not enforce a complementary distribution of pronouns and anaphors— as they do not in many versions of the Binding Theory, including Chomsky (1986a), Dalrymple (1993) and Koster and Reuland (1991) — then a wider distribution of PRO is incorrectly predicted, as is the case if Conditions A and B are not universally in complementary distribution. Moreover, if the PRO Theorem required that PRO not be governed, the Visibility Condition (naively construed under its various guises) would make it problematic to assign a thematic role to PRO—which after all is PRO’s raison d’être. In addition, Kayne (1991) has argued that PRO is in fact required to be governed, rather than always un Governed as the PRO theorem predicts. Finally, the GB arguments concerning the distribution and interpretation of PRO depend on theoretical tools which in the Minimalist Program are no longer available, including the Projection Principle, D-structure and S-structure, the notion of government, and the Binding Theory itself (bringing the PRO theorem into even greater doubt). Therefore there are both conceptual and empirical reasons to investigate Control Theory along other lines.

1.3.2 Case Theoretic Approaches: Null Case

Within the Minimalist Program, Lasnik (1993), as well as Chomsky and Lasnik (1991), Bošković (1995) and Martin (1996), argued that PRO must be Case-marked with “null” Case, where null Case is assigned by “certain instances of non-finite Tense.” Thus, without invoking the questionable “PRO theorem,” the distribution of PRO can be restricted appropriately.

Chomsky and Lasnik (1991) give several arguments in support of the idea that Case is assigned to PRO. The core examples used to make this point are the following from Chomsky and Lasnik (1991:p.118 in Chomsky 1995; bracketing added):

(11) a. it is rare [for it to strike John [that the problems are insoluble]]
b. it is rare [for it to seem to John [that the problems are insoluble]]

(12) a. *We want [John to strike t [that the problems are insoluble]]
b. *We want [John to seem to t [that the problems are insoluble]]
c. *We want [PRO to strike t [that the problems are insoluble]]
d. *We want [PRO to seem to t [that the problems are insoluble]]

Chomsky and Lasnik note that the sentences in (12c-d) are ungrammatical, despite the fact that PRO is ungoverned in those examples, and heads a theta-marked chain. The only possible factor rendering (12c-d) ungrammatical is the same reason that (12a-b) is ungrammatical: a DP cannot move from a Case-checking position to check Case. Therefore, they conclude that PRO in (12c-d) is moving in order to check Case—the “null case” which PRO needs and only PRO can accept.

Note that this assumption of null case automatically solves the PRO’s Visibility Condition problems discussed in the last subsection. Also, it has been known for a long time that PRO cannot occur in any position which checks other kinds of Case such as nominative or accusative. This leads to an obvious difficulty with the “null case” hypothesis: in languages like Icelandic or Russian, secondary predicates or floated quantifiers which agree in Case with PRO do not morphologically show this “null case”, but instead show one of the ordinary Cases (as we will discuss in later chapters). One could possibly sidestep the question of why “null case” does not appear on the secondary predicates, but it remains a mystery how the secondary predicates get their Case other than through agreement with PRO.

However, there is a more important conceptual flaw with the Chomsky and Lasnik’s argument. It is known that overt DP’s cannot move from a Case-checking position to later check Case. Instead of saying the PRO is assigned a mysterious Case in and only in the exact positions in which it must be anyway, let us suppose that the element being
moved from the Case-checking position in (12) is identical to the DP coindexed with it which is assigned Case in the matrix clause in (12). In other words, assume the structures for (12c-d) are as follows, with the LF structure for (13a) in (14):

(13)  
   a. *We₁ want [t₁ to strike t₁ [that the problems are insoluble]]
   b. *We₁ want [t₁ to seem to t₁ [that the problems are insoluble]]

(14)

---

5With details of head movement suppressed.
In (13a&b), corresponding to (12c&d) respectively, we see that there is a successive cyclic Case-chain connecting the potential Case-checking positions in the embedded clause to the actual Case-checking position in the matrix clause, in much the same way that a successive cyclic wh-chain connects an indeterminate number of potential wh-checking positions and ends with an actual wh-checking position. Of course, it is now obvious why the sentences in (13) are ungrammatical: despite a actual Case-checking position lower in the chain, Case-movement continued to another Case-checking position in the matrix clause. This is a violation of the basic movement and feature-checking principles of Minimalism. Let us ignore for the moment the substantial empirical and conceptual problems with this idea—they will be dealt with fully in later chapters—and consider the relative simplicity and clarity of this explanation.

From this point of view, the “null case” analysis seems, in a sense, to be begging the question. It asserts that PRO is an invisible entity being assigned an invisible Case (assignable to nothing else) under obscure circumstances, and in addition the PRO, unlike all other entities, cannot be assigned any other kind of Case. It is not answered why “null Case” is unique in this way and for that matter why PRO is unique in this way — it is simply stipulated. In addition, the null case analysis does not supply a theory of the interpretation of PRO.

As we will see in the next section and in the other chapters of this thesis, there is a sense in which the work presented here is related to this proposal. To a first approximation, the null Case proposal is identical to saying that a “null” Case position is assigning no Case at all, and that position is only a stop for a DP to move up to the matrix clause for Case. This is the analysis given in O’Neil (1994) and in the rest of this thesis.

1.3.3 Other Directions

Other proposals concerning the distribution of PRO have been advanced. Usually, such work attempts to reduce Control Theory to Binding Theory as in Bouchard (1983), or to
potential controllers’ theta roles as in Nishigauchi (1984) and Williams (1994), although I will not describe those theories in any more detail here, since they depend on GB mechanisms which are questionable under the assumptions of the Minimalist Program.

In contrast, this thesis argues that PRO and Control Theory can be entirely replaced by the proper combination of independent syntactic and semantic mechanisms, as we will soon see.

1.4 Beyond Control

The basis of this work is an elementary conceptual observation of the Minimalist Program, one of whose leading ideas is the elimination of complex conditions on syntactic structure with simpler, local conditions on derivations. For reasons of simplicity and “virtual conceptual necessity” (Chomsky 1994), only the articulatory-perceptual interface (PF) and the conceptual-intensional interface (LF) are distinguished as levels at which syntactic constraints may hold. As discussed above, both ideas contrast sharply with other principles and parameters theories of syntax such as GB Theory, as presented in Chomsky (1981, 1982, 1986b), and others. The ideas of the Minimalist Program, although well-motivated, have a number of consequences which seem undesirable at first glance.

In particular, D-structure is dispensed with, and consequently any possibility of theta-role assignment at D-structure is eliminated. Given the nature of theta roles and theta assignment (inherently they are not feature-driven), it is to be expected that theta role assignment would be an LF interface condition. However, the nature of theta roles and their assignment, in this analysis, remains obscure—especially since the critical condition of Theta Theory, the Theta Criterion, is eliminated along with D-structure. Also problematic in the Minimalist Program is the Projection Principle. In addition, the reformulation of the Binding Theory presents certain difficulties in the Minimalist Program. Previously believed to apply at S-structure, like Theta Theory, one presumes that in the Minimalist Program it holds at LF, but as we shall see problems arise in adapting the Binding Theory
to the mechanisms found in the Minimalist Program.

Having observed the difficulties of adapting both Theta Theory and Binding Theory to the Minimalist Program, it is perhaps unsurprising to note further that, insofar as there is a coherent theory of Control in GB Theory, it cannot be carried over into the Minimalist Program. These three modules—Theta Theory, Binding Theory, and Control Theory—appear to be incompatible with the Minimalist Program at a fundamental conceptual level. This incompatibility is further reinforced when one considers the semantic nature of these three modules. None of them are involved in the core operations of the Minimalist Program: structure building, movement and feature checking. Rather, all three modules refer to the fundamentally semantic relations of argument-taking and referentiality. As a result of this, it is plausible to investigate how and where Theta Theory, Binding Theory and Control Theory work, since they (or something like them) is certainly required, and previous answers seem to be excluded as possibilities.

A fundamental insight is from Williams (1980), who demonstrates that Control Theory may be divided into two separate phenomena with two different clusters of properties: “obligatory” Control and “non-obligatory” or “optional” Control. Consider the following pairs of sentences and their grammaticality judgements:

(15) a. John wanted PRO to behave himself/*oneself
    b. John thought that it was time PRO to behave himself/oneself

(16) a. John asked PRO to see himself/*oneself in the mirror
    b. John asked how PRO to see himself/oneself in the mirror

(17) a. John told Mary PRO to wash herself/*himself/*themselves
    b. John told Mary that it was time PRO to wash herself/himself/themselves

(18) a. John’s sister wanted PRO to behave herself/*himself
    b. PRO to behave myself/himself/oneself would be wrong
The examples in (15-18a) — obligatory control — have a number of similarities which
distinguish them from (15-18b) — optional control. In the former examples, there is
always a single unique interpretation for PRO, while in the latter examples there is a
number of possible interpretations, including the “arbitrary PRO” interpretation. In
obligatory control the antecedent must c-command PRO, whereas c-command is not
necessary for optional control. Finally, optionally controlled PRO has a number of features
which make its interpretation similar to a pronoun’s: we find split antecedence, multiple
possible antecedents, and long-distance antecedence. None of these are possible for
obligatorily controlled PRO.

What then is the difference between obligatory and optional control? Although both
types of control constructions appear as “missing” subject of non-finite clauses, closer
investigation shows that the different types of control have different properties, indicating
that different parts of the grammar may be responsible for them. In addition, we have
seen that there is no current theory of Control which integrates an explanation of the
occurrence of PRO with an explanation of the interpretation of PRO. It would be best if
both the occurrence and the interpretation aspects of Control could be derived from the
same set of assumptions.

It is proposed in O’Neil (1994) that there is no need for a separate theory of Control
in the grammar. In that paper, a theory of theta assignment at LF was proposed which
allowed theta roles from more than one head to be assigned to the same Case-chain under
certain conditions. Such multiple assignment of theta roles is no longer automatically
disallowed, since the Theta Criterion and the Projection Principle, which ruled it out in
GB Theory, are not part of the Minimalist Program. This was used to explain the properties
of occurrence and interpretation of “obligatory control” in the following sentence’s
(somewhat schematic) LF structure.

(19) [ Jamie, [vp t, wants [t, to [vp t, leave]]]]
The successive-cyclic Case-chain \{Jamie, t₁, t₂, t₃, t₄\} is within range to be theta-marked at LF by two different verbs, *wants* and *leave*, and receives both theta roles. Note that the difference between this and a raising predicate is that the only theta role assigned in raising predicates in the embedded clause, while “control” verbs assign an additional theta role to the moved element.⁶ In a similar vein, the properties of both the occurrence and the interpretation of other cases of obligatory control—such as the complements of verbs like *try*, *promise*, and *persuade*—are derived from elementary principles of Case-checking and theta-marking in the Minimalist Program. It is shown that long Case-chains which have multiple theta roles assigned to them make PRO superfluous in cases of

---

⁶There are some similarities between the theory of Control presented in this thesis and the proposals for these data in HPSG (Pollard and Sag 1987, 1994) and LFG (Bresnan 1982). However, these similarities are so obscured by the differences between the formalism of the Minimalist Program and unification-based grammars that direct comparisons between this work and those would most often not be very enlightening.
“obligatory Control”. If PRO could be entirely dispensed with, Control Theory would be superfluous, simplifying the grammar. Chapter 2 of this thesis is an elaboration of that paper.

Such a theory would also give an informal explanation for the approximate correctness of the Minimum Distance Principle of Rosenbaum (1970).

(20) **Minimum Distance Principle (MDP)**

An infinitive complement of a predicate P selects as its controller the minimal c-commanding noun phrase in the functional complex of P.

From the point of the view of the theory sketched out in the preceeding paragraph, the explanation for the MDP is simple. As we have already argued, Case-checking is constrained by the Shortest Move Requirement. If so, then the so-called “controller” of a nonfinite subject is really the head of a Case-chain. Given the fact that the head of the Case-chain must have moved from the subject position of the nonfinite clause through a shortest move (or a series of them)—as defined by c-command—we can see that the MDP might be a relatively good rule of thumb. As we will see in chapter 2, the main exceptions to the MDP (e.g. subject control verbs like *promise*) violate the MDP, but in precisely the way predicted by the minimalist principles of Case-checking and movement.

However, O’Neil (1994) gave little consideration to the problem of “optional Control”. In this thesis I would like also to explore this question, and show that, indeed, Control Theory is superfluous and can be eliminated. Unlike obligatory control, I will argue that optional control can be explained by the Binding Theory — in other words, I will argue that PRO should be replaced by element like pro.

Although this thesis will not explicitly argue against pervious versions of Control Theory, within the Minimalist Program or outside of it, the conceptual and empirical arguments against previous versions of Theta Theory or Binding Theory would hold
equally well against Control Theory—it is simply implausible, given the structure of the grammar hypothesized in the Minimalist Program. In addition, it seems clear that no one has never proposed a satisfactorily unified syntactic Control Theory. In such cases in the history of generative grammar, we have seen that problems of coverage of the data and complexity of the proposed solution arise from the attempt to reify a cluster of properties into a single mechanism; the solution, when found, recognizes that the purported phenomenon arises from the “conspiratorial” interaction of several simpler modules. I hope that this serendipity will hold of the proposals in this thesis.

1.5 Roadmap of the Thesis

This chapter has given the background information needed for the remainder of the thesis: an introduction to the Minimalist Program, to the history of the Theory of Control, and to the approach to control which is taken in this thesis. The rest of this thesis describes the mechanisms which I propose to replace Control Theory. In chapter 2, I discuss a new proposal for Theta Theory, and argue that this new version of Theta Theory, along with independently motivated requirements of movement and feature checking in the Minimalist Program, provide a theoretical explanation for obligatory control. In chapter 3 I discuss a number of additional phenomena and show that these proposals have a wide empirical coverage in English, and chapter 4 applies these proposals to other Romance and Germanic languages. In chapter 5, I show a number of other facts which follow from these proposals, and I argue that the properties of optional control are derivable from the Binding Theory. The dissertation concludes with chapter 6.