Derivations & Evaluations. On the syntax of subjects and complementizers
Dekkers, J.R.M.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
0. Introduction

This book is about the syntactic behavior of subjects and complementizers, and about the properties of the model of grammar that should explain this behavior.

Complementizers are often deleted. From a cross-linguistic point of view, this is a robust tendency. However, languages differ with respect to the extent to which they delete complementizers. In some of them, complementizers are absent altogether. Other languages pronounce complementizers in a smaller or greater number of contexts. A parallel with phonology can easily be drawn. Languages tend to avoid codas. Again, this is a robust tendency, although some languages show a strict CV-syllabification, while others are more liberal. It seems that whenever codas are realized, some other phonological requirement interferes and prevails. Similarly, context- and language-dependent complementizer pronunciation can be blamed on the interference of other syntactic requirements.

In the early 1990s, Paul Smolensky and Alan Prince introduced Optimality Theory (OT, see Prince & Smolensky, 1993). This theory allowed them to view grammar in terms of the interaction of conflicting grammatical tendencies. An OT grammar consists of a set of simple statements about linguistic objects (constraints) to which linguistic objects must comply. Since these constraints are in conflict with each other, it is impossible for linguistic objects to satisfy all of them. Hence, constraints are violable. Smolensky and Prince propose that these violable constraints are ranked and that constraint violation is allowed as long as it is minimal. Violation of a lower-ranked constraint is only permitted if this leads to satisfaction of a higher-ranked constraint. Consequently, the higher the rank of a constraint in the individual language, the more contexts there are in which we can observe its effect: OT is a theory of markedness.

This means that the question of whether and in which contexts codas or complementizers are realized depends on the relative rank of the pertinent constraints. The fact that the distribution of codas and complementizers varies from language to language can thus be related to the differences in constraint ranking that determine the grammars of individual languages.

Also subjects exhibit a marked syntactic behavior, in the sense that they often do not behave on a par with objects and adjuncts. The contexts in which the syntax of subjects actually differs from that of objects and adjuncts is language-dependent; so-called subject-object asymmetries are not universal. Again, we seem to deal with a phenomenon that can (and arguably should) be explained in terms of constraint interaction.

Hence, an attempt to explain these and related aspects of the syntax of complementizers in OT terms seems warranted. In the first section of this chapter, the basic properties of OT will be introduced. Subsequently, in section 2, we illustrate the way in which OT can be applied to syntax with an analysis of subject-auxiliary inversion in English. Special attention will be paid to English do-support.
This analysis, based on a flexible approach to phrase structure, serves as a model for analyses of syntactic problems proposed in subsequent chapters. The third and final section presents an outline of the book.

1. Optimality Theory

This section introduces the basic properties of OT. We will focus on the set-up of OT grammar, on tableaux as graphic representations of the evaluation procedure, on "tied" constraints, and on the notion harmonic boundedness.

1.1. The set-up of OT grammar

An OT grammar can be schematically represented as in figure 1. The generator (Gen) is a function which gives a (potentially infinite) set of candidate analyses for each input. According to the principle of inclusiveness, Gen produces all those analyses of the input that "are admitted by very general considerations of structural well-formedness" (McCarthy & Prince, 1993), such as the universal properties of phrase structure. In the minimal case, inputs consist of simplex linguistic objects. In syntax, the minimal input is a set of words, while the candidate set is the set of structures (including sentences) that Gen can build from these words.

```
input —— Gen —— candidate set —— Eval —— optimal candidate
```

Figure 1

Candidate sets are evaluated with respect to a particular ranking of the constraint inventory Con. This output-oriented procedure is usually referred to as Eval. Each member of Con is a simple statement about the form of the output, possibly in relation to the input. It is often assumed that Con is universal (cf. Ellison, to appear, for an evaluation of this assumption).

Because of the conflict between constraints, any linguistic structure will violate at least some of the constraints. However, constraint violation need not lead to ungrammaticality, since constraints are violable and ranked; those structures that minimally violate full rankings are optimal, and by definition, grammatical. An output structure minimally violates a ranking if all alternative structures that have an equal or better score on the lowest-ranked constraint score worse on the ranking dominating this constraint. If the constraint inventory is universal, parametrization can be fully reduced to differences in constraint ranking.

1.2. Tableaux: ranking and re-ranking

Imagine the abstract constraint set Con = \{CON_1, CON_2, CON_3, CON_4\}, and the abstract candidate set \{A, B, C, D, E, F\} associated with the input I. Let us suppose that the members of this set incur the constraint violations specified in (1).
INTRODUCTION

A violates CON₁ (twice), CON₃ (twice), and CON₄ (once).
B violates CON₁ (twice), and CON₂ (once).
C violates CON₁ (once), CON₃ (once), CON₄ (twice).
D violates CON₁ (once), and CON₂ (once).
E violates CON₁ (once), and CON₃ (twice).
F violates CON₁ (once), CON₂ (once), and CON₃ (once).

Eval should decide which of the members of {A, ..., F} are optimal. This depends on the ranking of the members of Con. Let us consider the language L₁, in which CON₁ outranks CON₂, CON₂ outranks CON₃, and CON₃ outranks CON₄ (conventional notation: CON₁ » CON₂ » CON₃ » CON₄). The evaluation of {A, ..., F} in L₁ is given in tableau (2). In OT tableaux, the top row gives the constraint ranking from left to right. In subsequent rows, constraint violations are given for each output structure. Each star represents one violation. In tableau (2), candidate C is optimal because all candidates that have an equal or better score on CON₃ (i.e. B, D, and F) score worse on the ranking CON₁ » CON₂.

<table>
<thead>
<tr>
<th>Language L₁</th>
<th>CON₁</th>
<th>CON₂</th>
<th>CON₃</th>
<th>CON₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>**!</td>
<td>.</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>**!</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>*</td>
<td>*</td>
<td></td>
<td>**</td>
</tr>
<tr>
<td>D</td>
<td>*</td>
<td>*!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>*</td>
<td></td>
<td>**!</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>*</td>
<td>*!</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

It is easiest to locate optimal candidates by reading tableaux from left to right. For each column, starting with the leftmost constraint, we have to determine which candidates incur the lowest number of violations. These candidates proceed to the next constraint, while all others are eliminated from the evaluation. This procedure is repeated until only one candidate survives, or until there are no constraints left. In tableau (2), candidates A and B violate the highest-ranked constraint CON₁ twice, whereas the other four candidates do so once. Hence, the second violation of CON₁ incurred by A and B is fatal, and these candidates are excluded from further evaluation. Fatal violations are accompanied by exclamation marks; cells to the right of fatal violations are shaded because they are irrelevant for the evaluation. When we move on to CON₂, two of the remaining candidates (D and F) incur one violation, whereas the other two (C and E) incur none. Therefore, only C and E continue. Finally, candidate E is eliminated by CON₃, since it incurs one more violation than C. This means that candidate C is optimal, and by definition grammatical.

Alternative rankings of the four constraints in Con produce languages distinct from L₁. Let us associate the ranking CON₄ » CON₃ » CON₂ » CON₁ with the language L₂. The evaluation of {A, ..., F} in L₂ with respect to I proceeds as in tableau (3). In this tableau, only candidates A and C violate the highest-ranked
constraint $CON_4$. Hence, these two candidates are excluded from further competition. Only two of the remaining four candidates do not violate $CON_3$, viz. B and D. These two candidates continue, and survive the next constraint ($CON_2$), which is violated once by both of them. Finally, the lowest constraint in the ranking, $CON_1$, produces a winner. Candidate D violates this constraint once, whereas candidate B does so twice. Hence, candidate D is optimal and, by definition, grammatical. Notice that exclamation marks directly follow fatal violations. Candidate C, for instance, violates $CON_4$ twice. Since the first violation is fatal, it is followed by an exclamation mark. For candidate B, on the other hand, the second violation of $CON_1$ is fatal. Hence, the second asterisk in the cell is accompanied by an exclamation mark.

Thus, each of the twenty-four possible rankings of these four constraints leads to a distinct grammar. Since our candidate set $\{A, ..., F\}$ has only six members, two grammars $G_i$ and $G_j$ may select the same optimal candidate. This is in principle coincidental; other candidate sets may distinguish between $G_i$ and $G_j$.

1.3. Tableaux: tied constraints

In tableaux (2) and (3), all constraints are ranked with respect to each other. However, in subsequent chapters, we assume that constraints can be “in a tie”. If the constraints $CON_i$ and $CON_j$ are in a tie (notation: $CON_i \Leftrightarrow CON_j$), the mutual ranking of $CON_i$ and $CON_j$ is not specified. Let us return to the constraint set $CON$ and the candidate set $\{A,...,F\}$ of the previous subsection. Imagine a language $L_3$ characterized by the ranking $CON_2 \gg CON_3 \Leftrightarrow CON_4 \gg CON_1$ ($CON_3$ and $CON_4$ are in a tie). The tableau associated with the evaluation of the candidate set $\{A,...,F\}$ is given under (4). The dotted line marks the tie.

<table>
<thead>
<tr>
<th>Language $L_2$</th>
<th>$CON_4$</th>
<th>$CON_3$</th>
<th>$CON_2$</th>
<th>$CON_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>*!</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>*</td>
<td>**!</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td><em>!</em></td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>*!</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td><em>!</em></td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Furthermore, (i) all constraints that outrank $CON_i$ outrank $CON_j$, (ii) all constraints that outrank $CON_i$ outrank $CON_j$, (iii) all constraints that are outranked by $CON_i$ are outranked by $CON_j$, and (iv) all constraints that are outranked by $CON_i$ are outranked by $CON_j$. 
INTRODUCTION

(4)

<table>
<thead>
<tr>
<th>Language $L_3$</th>
<th>$\text{CON}_2$</th>
<th>$\text{CON}_3$ &amp; $\text{CON}_4$</th>
<th>$\text{CON}_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>B</td>
<td>*</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>C</td>
<td>*</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>D</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>E</td>
<td>**</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>F</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

There are two ways to interpret tableau (4). According to the first interpretation, the tied constraints $\text{CON}_3$ and $\text{CON}_4$ behave like one complex constraint. This would mean that the violations of these constraints incurred by each candidate are added up. The evaluation of candidate set \{A, ..., F\} in accordance with this interpretation is given in tableau (5). There are three candidates (A, C, E) that do not violate the highest-ranked constraint $\text{CON}_2$. These candidates are evaluated with respect to the complex constraint $\text{CON}_3 \& \text{CON}_4$. Since candidates A and C incur one more violation than candidate E, the latter candidate will be optimal.

(5)

<table>
<thead>
<tr>
<th>Language $L_3$</th>
<th>$\text{CON}_2$</th>
<th>$\text{CON}_3 &amp; \text{CON}_4$</th>
<th>$\text{CON}_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>***!</td>
<td>**</td>
</tr>
<tr>
<td>B</td>
<td>*</td>
<td>***!</td>
<td>**</td>
</tr>
<tr>
<td>C</td>
<td>*</td>
<td>***!</td>
<td>*</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>***!</td>
<td>*</td>
</tr>
<tr>
<td>E</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Alternatively, if constraints are in a tie, any fully specified ranking which does not contradict the original underspecified ranking may be used during the actual evaluation. The ranking $\text{CON}_2 \gg \text{CON}_3 \ll \text{CON}_4 \gg \text{CON}_1$ is compatible with both $\text{CON}_2 \gg \text{CON}_3 \gg \text{CON}_4 \gg \text{CON}_1$ and $\text{CON}_2 \gg \text{CON}_4 \gg \text{CON}_3 \gg \text{CON}_1$. This would mean that $L_3$ is a mix of $L_4$ (characterized by $\text{CON}_2 \gg \text{CON}_3 \gg \text{CON}_4 \gg \text{CON}_1$) and $L_5$ (characterized by $\text{CON}_2 \gg \text{CON}_4 \gg \text{CON}_3 \gg \text{CON}_1$). In other words, all candidates that are optimal in tableaux (6) and (7) (candidates C and E) are optimal in tableau (4).
Following Pesetsky (1997, 1998) and Broekhuis & Dekkers (to appear), we adopt the second interpretation, which will play an important role throughout this book:

(8) **Constraint tie (Pesetsky, 1998):**  
The output of a set of tied constraints is the union of the outputs of every possible ranking of those constraints.

The following conventions are used in tableaux containing two tied constraints (borrowed from Broekhuis & Dekkers, to appear):

(9) **Conventions:**
   a. $*$ = fatal violation if the tie is read from left to right.
   b. $<*$ = fatal violation if the tie is read from right to left.
   c. $*!$ = fatal violation in both directions.

   *Note that only candidates incurring fatal violations in both directions of a tie lose: $<*$ & $*$ = $*!$*

Let us use these conventions in tableau (4), repeated here under (10). There are three candidates that do not violate CON$_2$, viz. A, C, and E. If the tie between CON$_3$ and CON$_4$ is read from left to right, the second violation of CON$_3$ incurred by A and E (marked by the sign $>$) is fatal. Consequently, candidate C is optimal. If the tie is read from right to left, the first violation of CON$_4$ incurred by A and C (marked by the sign $<$) is fatal, which means that candidate E is optimal. Hence, the two readings together produce both C and E as optimal candidates (the only two
candidates that are not marked with an exclamation mark or with two pointed brackets).

\[(10)\]

<table>
<thead>
<tr>
<th>Language (L_3)</th>
<th>(\text{CON}_2)</th>
<th>(\text{CON}_3)</th>
<th>(\text{CON}_4)</th>
<th>(\text{CON}_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td><strong>&gt;</strong></td>
<td><strong>&lt;</strong></td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>*!</td>
<td></td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>*</td>
<td><em>&lt;</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td><strong>&gt;</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>*!</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.4. Harmonically bound candidates

A language-independent tableaux in which constraint violations are marked for the members of the by now familiar candidate set \{A, ..., F\} is given in (11). Often, the question of whether a specific candidate is optimal depends on the constraint ranking characterizing the language in question. Sometimes, however, a candidate will not be able to win under any ranking. This is what happens if the candidate is harmonically bound by some other candidate.

\[(11)\]

<table>
<thead>
<tr>
<th>(\text{CON}_1)</th>
<th>(\text{CON}_2)</th>
<th>(\text{CON}_3)</th>
<th>(\text{CON}_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>**</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>**</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>C</td>
<td>*</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>D</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>*</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>F</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

A candidate \(C_1\) is harmonically bound by some candidate \(C_2\) iff \(C_1\) incurs all constraint violations \(C_2\) does, but not vice versa. In that case, \(C_1\) will lose under any ranking. In tableau (11), candidate E harmonically binds candidate A: candidate E violates \(\text{CON}_1\) once and \(\text{CON}_3\) twice, whereas candidate A also incurs a second violation of \(\text{CON}_1\) as well as a violation of \(\text{CON}_4\). As a result, candidate A will lose under any ranking. The same holds for candidates B and F, which are both harmonically bound by candidate D.

2. OT applied to syntax

In this section, we illustrate how OT can be applied to syntax by means of a concrete analysis of English subject-auxiliary inversion and do-support. The analysis
is adapted from Grimshaw (1997), and presupposes her Extended X-bar Theory (Grimshaw, 1991, 1997).

2.1. Subject-auxiliary inversion

In the interrogative (12), the *wh*-element is parsed in its left-peripheral scope position and the auxiliary precedes the subject. In declaratives, on the other hand, no such preposing takes place, and subject-auxiliary inversion is illicit, as is shown in (13).

(12) Which books will he read?

(13) a. He will read these books.
    b. *Will he read these books.

Let us assume that the question of whether *wh*-movement takes place depends on the mutual ranking of the constraints PARSE-*wh* and STAY defined in (14). PARSE-*wh* is violated for each *wh*-element that does not appear in its scope position, while STAY is violated for each occurrence of a movement trace.\(^2\) Clearly, these two constraints conflict. If a *wh*-element moves to its scope position to satisfy PARSE-*wh*, STAY is violated. If, on the other hand, the *wh*-element stays in situ, STAY is satisfied, but PARSE-*wh* is violated. The fact that *wh*-elements are fronted in English suggests that PARSE-*wh* outranks STAY.

(14) a. PARSE-*wh*: Parse *wh*-elements in their scope position.
    b. STAY: Traces are prohibited.

Let us assume for the moment that PARSE-*wh* requires that *wh*-elements appear in a left-peripheral specifier position. In the case of (12), SpecIP cannot host *which books*, since the subject occupies this position. Hence, CP is needed to provide us with the necessary specifier. Let us suppose that *which books* moves to SpecCP. All things being equal, this leads to the structure in (15).

(15) \[
\text{LCP} \ [\text{IP} \ [\text{CP} \ [\text{IP} \ [\text{SpecCP} \ [\text{IP} \ \text{he will} \ [\text{VP} \ \text{read} \ t_1]]]]]]}
\]

This structure contains an empty head. Grimshaw (1997) argues that empty heads are marked. She introduces the constraint in (16), which requires that heads be realized. Heads are realized whenever they are occupied by syntactically or semantically substantive material, such as (traces of) phonetically realized material or, as is the case for I, temporal information (see section 2.2 below).

\(^2\) PARSE-*wh* corresponds to Grimshaw’s OP-SPEC, given in (i). Our choice for the term PARSE-*wh* will become clear in chapter 2. The definition of this constraint will be revised in that chapter. For now, it suffices to assume that any left-peripheral specifier qualifies as a scope position. Notice that this makes the definition of PARSE-*wh* more restrictive than that of OP-SPEC, which allows operators to appear in any specifier position. Nothing depends on this. See also footnote 5.

(i) OPERATOR IN SPECIFIER (OP-SPEC):
    Syntactic operators must be in specifier position.
(16) **OBLIGATORY HEADS (OB-HD):** A projection has a head.

The auxiliary is moved to C to satisfy OB-HD. Since movement is costly, this entails a violation of STAY. The fact that movement nevertheless takes place should be attributed to the fact that OB-HD outranks STAY in English. Since PARSE-\(w\) also outranks STAY, the English ranking is as given in (17).\(^3\)

(17) **English ranking:** \(\text{PARSE-}w\) \(\gg\) OB-HD \(\gg\) STAY

The evaluation of English interrogatives is given in tableau (18). Candidate (18c) is optimal, since it is the only structure that violates neither PARSE-\(w\), nor OB-HD.

(18)

<table>
<thead>
<tr>
<th>English</th>
<th>P-(w)</th>
<th>OB-HD</th>
<th>STAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. ([\text{IP he [I will]} [\text{VP read which books}]])</td>
<td>!*\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. ([\text{CP which books, [C }\emptyset\text{]} [\text{IP he [I will]} [\text{VP read }t_1]]])</td>
<td></td>
<td>!*\</td>
<td>*</td>
</tr>
<tr>
<td>c. ([\text{CP which books, [C will]} [\text{IP he [I }t_j\text{]} [\text{VP read }t_1]]])</td>
<td>!*\</td>
<td>!*\</td>
<td>*</td>
</tr>
<tr>
<td>d. ([\text{CP [C will]} [\text{IP he [I }t_1\text{]} [\text{VP read which books}]]])</td>
<td>!*\</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

In (19), it is illustrated that the presence of CP in English declarative clauses does not serve any purpose; all candidates vacuously satisfy PARSE-\(w\). Therefore, the presence of CP will always entail additional violations of OB-HD or STAY.

(19)

<table>
<thead>
<tr>
<th>English</th>
<th>P-(w)</th>
<th>OB-HD</th>
<th>STAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. ([\text{IP he [I will]} [\text{VP read these books}]])</td>
<td>!*\</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>b. ([\text{CP [C }\emptyset\text{]} [\text{IP he will [VP read these books]}]])</td>
<td>!*\</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. ([\text{CP [C will]} [\text{IP he [I }t_1\text{]} [\text{VP read these books}]]])</td>
<td>!*\</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Thus, CP is projected if this is required by the constraint ranking. Note that alternative rankings lead to different functional structure. If STAY outranks PARSE-\(w\), no overt \(w/z\)-movement will apply, and CP will not be projected. This seems to be true for a \(w/z\)-in-situ language like Chinese, as is illustrated in (20), taken from Huang (1982).

(20) \(\text{Ni kanjian-le shei?}\)
    you see-PERF who?

---

\(^3\) For an argument in favor of the ranking \(\text{OP-SPEC (PARSE-}w\) \(\gg\) OB-HD, the reader is referred to Grimshaw (1997: 396).
At the same time, constraint ranking determines how projected functional structure is realized. For instance, in languages which rank STAY higher than OB-HD, I-to-C movement does not take place. These languages may resort to other means to prevent a violation of OB-HD. A Canadian French example, taken from Lefebvre (1979), is given in (21). In this example, the complementizer *que* is inserted in C, which satisfies both OB-HD and STAY.\(^4\)

(21) a. Quand *que* Marie viendra?
   when *that* Marie come-FUT
   ‘When will Marie come?’
   b. [CP quand [C *que*] [IP Marie viendra]]

2.2. *Do*-support: a subject-object asymmetry

Let us return to English subject-auxiliary inversion. When a *wh*-phrase other than a subject is fronted, the subject and the auxiliary are clearly inverted. This follows if head movement is caused by the need to fill the head of the CP created to host the *wh*-phrase. If no *wh*-movement applies, CP is absent, and no inversion is attested. Hence, the declarative clause in (13a) and the interrogative clause in (12) should be parsed as in (22a) and (22b), respectively.

(22) a. [IP he [I will] [VP read these books]]
   b. [CP which books, [C willj] [IP he [I t] [VP read t]]]

However, it is less clear how to parse (23). In this example, the sentence-initial *wh*-phrase is a subject. The observed order of constituents matches that in (22a) (because the clause starts with the sequence subject-auxiliary) as well as that in (22b) (because the first two elements in the clause are the *wh*-element and the auxiliary).

(23) Who will read these books?

The structures in (24) show that the linear order in (23) is compatible with both an IP and a CP analysis. In (24a), the *wh*-subject is parsed in its Case position, and the auxiliary has remained in I. In (24b), on the other hand, the *wh*-subject has moved from SpecIP to SpecCP. This movement is string-vacuous because the auxiliary has undergone I-to-C raising.

(24) a. [IP who [I will] [VP read these books]]
   b. [CP who, [C willj] [IP tj [I t] [VP read these books]]]

Grimshaw argues in favor of the analysis given in (24a). If *wh*-elements are in their scope position as soon as they appear in a (left-peripheral) specifier

\(^4\) The question of whether this insertion is actually permitted will depend on the rank of a constraint like TEL (see Pesetsky, 1997, 1998; Broekhuis & Dekkers, to appear), which will be introduced in the next chapter. This constraint prohibits the pronunciation of function words such as complementizers.
position, then both structures in (24) satisfy PARSE-wh. This means that the presence of CP will always lead to unnecessary violations of (at least) STAY. Tableau (25) (= (24a)) shows that candidate (a) is the only structure that does not violate any of the three relevant constraints.\(^5\) Hence, it is optimal.

\[\text{(25)}\]

<table>
<thead>
<tr>
<th></th>
<th>P-wh</th>
<th>OB-HD</th>
<th>STAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. [IP who [(1) will] [VP read these books]]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. [CP who, [C (\emptyset)] [IP (t_1) [(1) will] [VP read these books]]]</td>
<td></td>
<td>!</td>
<td></td>
</tr>
</tbody>
</table>
| c. [CP who, [C will,] [IP \(t_1\) [\(1\) \(t_1\)] [VP read these books]]] |  | ! | !*
| d. [CP [C will,] [IP who [\(1\) \(t_1\)] [VP read these books]]] | ! | * | *|

So far, the motivation for the absence of CP in the presence of a wh-subject has been theory-internal. However, let us turn to the examples in (26). If a declarative main clause lacks an auxiliary, its interrogative counterpart will contain do if the wh-phrase is not a subject.

\[\text{(26)}\]

a. He read these books.
b. Which books did he read.
b'. *Which books he read?
c. Who read these books?
c'. *Who did read these books?

Let us assume that the structures associated with the examples in (26a-b) correspond to those in (22a-b). This means that (26a) should be analyzed as an IP. The structure is given in (27a). In (26b), on the other hand, the order auxiliary-subject signals the presence of CP. This leads to the structure in (27b), in which the wh-object has moved to SpecCP, while the auxiliary did occupies C. When we turn to (26c), the three possible structures in (27c-c") are compatible with the attested word order. However, (27c'-c") are unlikely parses as long as we want to maintain that interrogative main clauses in English do not allow empty Cs or Cs filled by a lexical verb. If (26c) is analyzed as in (27c), on the other hand, the presence of did can be directly linked to the presence of CP.

\[\text{(27)}\]

a. [IP he [\(1\) [VP read these books]]]
b. [CP which book, [C did,] [IP he [\(1\) \(t_1\)] [VP read these books]]]
c. [IP who [\(1\) [VP read these books]]]
c'. [CP who, [C \(\emptyset\)] [IP \(t_1\) [\(1\) [VP read these books]]]
c". [CP who, [C read,] [IP \(t_1\) [\(1\) \(t_1\)] [VP \(t_1\) these books]]]

\(^5\) Notice that candidate (25d) does not violate Grimshaw’s (1997) original constraint OP-SPEC formulated in footnote 2. However, if we substitute OP-SPEC for PARSE-wh in tableau (25), the candidate in question will still be suboptimal, due to its violation of STAY.
Before we go and examine (26b-c) in further detail, let us first consider the position of the verb in (26a). A classical paradigm is given in (28) and (29). In French, VP adverbs follow rather than precede the finite lexical verb, while the opposite holds for English.

(28) a. Jean embrasse souvent Marie.
    Jean kisses often Marie
    'Jean often kisses Marie.'
    b. *Jean souvent embrasse Marie.

(29) a. *John kisses often Mary,
    b. John often kisses Mary.

This suggests that the finite lexical verb is in a higher position in French than it is in English. Let us hypothesize, in accordance with standard assumptions, that in English, the verb stays in its base position, while it moves to I in French:

(30) a. [IP [T [I Jean [I embrasse]]] [VP [t [IVP souvent t Marie]]]]
    b. [IP [T [I John [I ]] [VP [t oft en [IVP kisses Mary]]]]]

The structures in (30) suggests that a constraint requiring that the inflected verb be parsed in I interacts with STAY. Let us assume that verbs move to T to satisfy PARSE-T, which is violated for each finite V that is not parsed in a position containing Tense features (i.e. I). When PARSE-T outranks STAY, the inflected verb will raise to I, as in French. The absence of V-to-I movement in English can be reduced to the ranking STAY >> PARSE-T. Note that V-to-I movement is not needed to satisfy OB-HD because I contains temporal information and therefore counts as a realized head (section 2.1 above).

(31) PARSE-T: Pronounce inflected V in a position containing Tense features.

Let us return to (27a), repeated in (32a). The verb is in its base-position inside the VP. Alternative structures are given in (32b-c). Structure (32b), in which the verb has moved to I, is eliminated by STAY, which outranks PARSE-T. Next to V-to-I movement, there is the possibility of pronouncing did in I, as in (32c) (containing non-emphatic did). If did is base-generated in I, this does not entail a violation of STAY.

(32) a. [IP [T [I he [I ]] [VP [t [IVP read these books]]]]]
    b. [IP [T [I he [I read]] [VP [t [IVP these books]]]]]
    c. [IP [T [I did [I]] [VP [t [IVP read these books]]]]]

However, (32c) does not surface in English. Let us assume that this is due to an economy constraint disallowing words.6

6 Alternatively, do-support could be banned by a faithfulness constraint prohibiting syntactic elements that are not contained in the input. However, not only does do-support involve do-insertion, it also changes the finite lexical verb into an infinitive. This raises intricate questions
NOWORD: Words are prohibited.

This constraint is violated for each word contained in a given structure. Clearly, NOWORD is a violable constraint, since winning candidates always contain words (cf. Ackema & Neeleman, to appear). For now, let us assume that words essential to the semantic interpretation of the sentence cannot be omitted. We will return to this point in detail in chapter 3. Like STAY, NOWORD outranks PARSE-T, witness the ungrammaticality of (32c). The evaluation of the structures in (32) is given in tableau (34). V-to-I raising (candidate (34b)) is excluded due to the high rank of STAY, while NOWORD eliminates candidate (34c) containing did in I, which incurs one more violation than candidate (34a) does. Hence, candidate (34a) is optimal. The ranking STAY >> NOWORD is irrelevant here, but will be crucial below.

Main clause object interrogatives are evaluated in tableau (35). The ranking we have established for English is PARSE-w/h >> OB-HD >> {STAY, NOWORD} >> PARSE-T. The highest-ranked of these constraints, PARSE-w/h, eliminates candidates (35a-e) because the w/h-phrase does not appear at the left edge of the clause. Candidates (35f-h), on the other hand, incur a fatal violation of OB-HD because they contain an empty C. Notice that although I is not occupied by (a trace of) a phonetically realized head in (35a) and (35f), this does not lead to (additional) violation(s) of OB-HD because I contains temporal information. This leaves us with candidates (35i-j). In (35j), did has moved to C, while in (35i), the lexical verb has been so. Because the lexical verb originates from inside the VP and did from I, (35i) contains one more trace than (35j). Therefore, (35j) is optimal under the ranking STAY >> NOWORD, and correctly predicted to be grammatical. If we had assumed the opposite ranking, (35i) would have been optimal because this candidate contains fewer words than (35j) does.

about the nature of the relation between the input and the output. We will argue in subsequent chapters that syntactic elements can freely be inserted as long as the distribution of formal features is kept constant. When using *WORD instead of a faithfulness constraint, this allows us to circumvent input-output problems. See chapter 2 (in particular section 4.2) for further discussion.

Cf. Grimshaw (1997: 382), who argues that IP is absent in English clauses that do not contain an auxiliary. The special status of IP will be further discussed in chapter 2.
In main clause subject interrogatives, the same ranking selects a candidate lacking do-support. In tableau (36), candidates (d) and (e) violate PARSE-wh because the wh-element does not appear in left-peripheral position. All other candidates satisfy this constraint, irrespective of whether the w/z-phrase occupies SpecIP or SpecCP. Candidates (35f-h), in which C is empty, incur fatal violations of OB-HD. Among the remaining five candidates, there are only two that do not violate STAY, i.e. (36a) and (36c). Since (36c) incurs five violations of NOWORD, and (36a) only four, (36a) is optimal.

<table>
<thead>
<tr>
<th>English</th>
<th>P-wh</th>
<th>OB-HD</th>
<th>STAY</th>
<th>NOWORD</th>
<th>P-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. [IP he [t] [VP read which books]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. [IP he [t read]] [VP t which books]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. [IP he [t did] [VP read which books]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. [CP which read] [IP he [t]] [VP t which books]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. [CP which did] [IP he [t]] [VP read which books]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f. [CP which books] [IP he [t]] [VP read t]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g. [CP which books] [IP he [t]] [VP read t]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h. [CP which books] [IP he [t]] [VP read t]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i. [CP which books] [IP he [t]] [VP read t]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j. [CP which books] [IP he [t]] [VP read t]]</td>
<td>*!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In main clause subject interrogatives, the same ranking selects a candidate lacking do-support. In tableau (36), candidates (d) and (e) violate PARSE-wh because the wh-element does not appear in left-peripheral position. All other candidates satisfy this constraint, irrespective of whether the w/z-phrase occupies SpecIP or SpecCP. Candidates (35f-h), in which C is empty, incur fatal violations of OB-HD. Among the remaining five candidates, there are only two that do not violate STAY, i.e. (36a) and (36c). Since (36c) incurs five violations of NOWORD, and (36a) only four, (36a) is optimal.
In sum, the subject-object asymmetry in (26) can be reduced to a difference in clause size, as in (37).

\[(37)\]

\[a. \ [CP \text{ which book; } [\text{CP did} \ [\text{IP he } [I \text{ read } t_i] \ [\text{VP read } t_i]]]]\]

\[b. \ [\text{IP who } [I \text{ read } t_i] \ [\text{VP t_i these books}]]\]

This is because our constraint ranking requires that main clause interrogatives in English be analyzed as IPs when they are introduced by a wh-subject, and as CPs when they start with some other wh-element. At the same time, the ranking relates the possibility of do-support in interrogatives to the presence of CP.

2.3. Extended X-bar Theory

In this chapter, we have implicitly adopted an approach to phrase structure along the lines of Grimshaw’s Extended X-bar Theory (1991, 1997). According to Grimshaw, functional projections do not select their lexical complements. Rather, functional projections are extended projections of lexical heads.

In standard X-bar Theory, each projection has exactly one head. In the tree diagram in (38), NP and N’ are headed by N, DP and D’ by D, VP and V’ by V, IP and I’ by I, and CP and C’ by C. Other relations cannot be captured in terms of projection alone. Some of these relations could be cases of selection. For instance, it could be that C selects IP. Other relations can only be captured in terms of
domination, e.g., N is dominated by DP. However, although N is also dominated by IP, the relation with this element is less intimate because a lexical projection (VP) is interfering. According to Grimshaw, DP is an extended projection of N, while IP is not.

Grimshaw (1991) makes a distinction between the notions perfect head/projection and extended head/projection. The notion perfect head and perfect projection are synonymous with the notions head and projection in standard X-bar Theory. In (38), IP is a perfect projection of I, and not of V; N is a perfect head of N, and not of DP, etc. The notions extended head and extended projection are defined in (39): an extended head must be dominated by its extended projection (clause (a)), an extended head must share its categorial features with its extended projection (clause (b)), no projections that are not extended projections of the head in question may intervene between the extended head and the extended projection (clause (c)), and each extended projection has only one lexical extended head (clause (d)).

\[
\begin{align*}
\text{(38)} & \\
\text{CP} & \\
\text{C} & \\
\text{IP} & \\
\text{I'} & \\
\text{VP} & \\
\text{V'} & \\
\text{V} & \\
\text{DP} & \\
\text{D'} & \\
\text{D} & \\
\text{NP} & \\
\text{N'} & \\
\text{N} & 
\end{align*}
\]

Grimshaw (1991) makes a distinction between the notions perfect head/projection and extended head/projection. The notion perfect head and perfect projection are synonymous with the notions head and projection in standard X-bar Theory. In (38), IP is a perfect projection of I, and not of V; N is a perfect head of N, and not of DP, etc. The notions extended head and extended projection are defined in (39): an extended head must be dominated by its extended projection (clause (a)), an extended head must share its categorial features with its extended projection (clause (b)), no projections that are not extended projections of the head in question may intervene between the extended head and the extended projection (clause (c)), and each extended projection has only one lexical extended head (clause (d)).

\[
\begin{align*}
\text{(39)} & \\
\alpha & \text{ is the extended head of } \beta, \text{ and } \beta \text{ is an extended projection of } \alpha \text{ iff (a), (b), (c) and (d) hold:} \\
a. & \beta \text{ dominates } \alpha. \\
b. & \alpha \text{ and } \beta \text{ have identical categorial features.} \\
c. & \text{if } \gamma \text{ dominates } \alpha \text{ and } \beta \text{ dominates } \gamma, \gamma \text{ is an extended projection of } \alpha. \\
d. & \text{if } \delta \text{ is an extended projection of } \alpha \text{ and } \delta \text{ is lexical, } \delta \text{ is a perfect projection of } \alpha.
\end{align*}
\]
Let us illustrate this with some examples. In (38), CP, C', IP, I', VP, and V' are extended projections of V, and V is the extended head of these projections. Similarly, I is the extended head of I', IP, C', and CP, and these projections are extended projections of I. However, IP is not an extended projection of D or N, both because IP is verbal while D and N are nominal, and because this would mean that IP would have two lexical extended heads (V and N).

Grimshaw (1997) argues that OT syntax associates the optimal extended projection with a given lexical structure. Functional structure is freely added by Gen and Eval determines how much of this structure is really needed on top of a given lexical projection, as in the analysis of subject-auxiliary inversion presented above. In principle, functional projections have no intrinsic semantic or syntactic content; they are dependent on the information on the lexical head of the extended projection. Syntactic information associated with the lexical head is transmitted to the functional domain through movement (or percolation).

Extended X-bar Theory also has repercussions for the theory of selection. Grimshaw (1991) distinguishes between syntactic (c-selection) and semantic selection (s-selection). Lexical heads c-select for syntactic category, which means that a verb does not c-select an IP, but rather a verbal extended projection. As a result, c-selection will never prefer an IP to a CP or a VP. Again, the actual choice between different instantiations of verbal extended projections will be made by Eval. As we will see in subsequent chapters, this makes it possible, for instance, to reduce the presence or absence of complementizers across languages (see the examples in (40)) to the presence or absence of CP.

(40) a. **Obligatory complementizers (French):**
   Je pense *(que)* le Président de la République a déguisé la vérité.
   'I think that the French president has covered up the truth.'

b. **Optional complementizers (English):**
   I think (that) the President has covered up the truth.

c. **No complementizers (Chinese):**
   Paul juede zongtong sahuang-le.
   'Paul thinks the President has lied.'

S-selection, on the other hand, is for semantic type. In chapters 4 and 5, we will examine this type of selection in more detail.

### 3. Outline of the book

In this book, we examine a variety of subject-object asymmetries and other aspects of the syntax of subjects and complementizers which will all be linked to clause size, along the lines of the analysis of *do*-support presented above. Economy constraints play a central role throughout.

The book consists of five chapters. In chapter 2, we will present our model of syntax in more detail. The starting-point will be Chomsky’s (1995) Minimalist
Program. It will be argued that the syntactic system developed within this program should not be considered an alternative to OT syntax, since the two theories are largely complementary. While OT is output-oriented, the Minimalist Program focuses on derivations. This will ultimately lead us to adopt a revised version of Broekhuis & Dekkers’ (to appear) Derivations & Evaluations framework, an OT syntax in which Chomsky’s syntactic system serves as a generator.

Chapter 3 focuses on the left periphery of the relative clause. We will examine the issue of deletion of syntactic material in general, and of complementizers and relative pronouns in particular. The size of relative clauses gets special attention. Subject-object asymmetries in this clause type are related to the presence or absence of CP, just like do-support in English main clause interrogatives.

Chapter 4 will be entirely devoted to subject-object asymmetries in the context of long wh-movement. The first part of the chapter consists of an overview of the ways in which the so-called Complementizer-trace Phenomenon has been accounted for since the early 1970s. In the second part, we will propose an alternative analysis, again based on variable clause size, i.e. on the presence or absence of CP.

In chapter 5, we will examine the left-periphery of French Stylistic Inversion in several clause types. In this construction, the subject appears in postverbal position, whereas in the unmarked case, subjects precede the verbal complex in French. The analysis put forth shows that variable clause size allows us to explain more than subject-object asymmetries. Whereas Stylistic Inversion is traditionally analyzed as CP, we will argue in favor of a uniform IP analysis. In general, Stylistic Inversion clauses have an SVO counterpart that should be analyzed as CP. It will be argued that a number of differences between Stylistic Inversion and SVO clauses can be reduced to this difference in clause structure.

Chapter 6 summarizes the main conclusions drawn in this book.