Linguistik Aktuell Linguistics Today 225

# Rethinking Syntactocentrism

## Architectural issues and case studies at the syntax-pragmatics interface

Andreas Trotzke

John Benjamins Publishing Company

# Rethinking Syntactocentrism

## Architectural issues and case studies at the syntax-pragmatics interface

Andreas Trotzke
University of Konstanz

John Benjamins Publishing Company
Amsterdam / Philadelphia

The paper used in this publication meets the minimum requirements of the American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

DOI 10.1075/la.225

Cataloging-in-Publication Data available from Library of Congress:
LCCN 2015024565 (PRINT) / 2015027720 (E-BOOK)

ISBN 978 90 272 5708 6 (HB)
ISBN 978 90 272 6806 8 (E-BOOK)



John Benjamins Publishing Co. · https://benjamins.com

# Rethinking Syntactocentrism

*Linguistik Aktuell/Linguistics Today (LA)*
ISSN 0166-0829

*Linguistik Aktuell/Linguistics Today (LA)* provides a platform for original monograph studies into synchronic and diachronic linguistics. Studies in LA confront empirical and theoretical problems as these are currently discussed in syntax, semantics, morphology, phonology, and systematic pragmatics with the aim to establish robust empirical generalizations within a universalistic perspective.

For an overview of all books published in this series, please see
*http://benjamins.com/catalog/la*



**Volume 225**

Rethinking Syntactocentrism
Architectural issues and case studies at the syntax-pragmatics interface
by Andreas Trotzke

# Table of contents

CHAPTER 1

# Introduction

> [W]hat is called for is an open-mindedness to insights from whatever quarter [...] and a joint commitment to fight fair in the interests of deeper understanding. To my mind, that's what the game of science is about. (Jackendoff 2002: xiii)

> [T]he syntactocentric model [...] was explicitly only an assumption, which quickly hardened into dogma and then became part of the unstated background. (Jackendoff 2003: 659)

The linguist Ray Jackendoff never tires to call for open-mindedness and fairness in the heterogeneous area of research on the mental foundations of language. One of the approaches within this field claims that syntax, as regarded in linguistics à la Noam Chomsky, plays the central role in modeling the mental architecture of the human language faculty. According to Jackendoff, this conception has 'hardened into dogma' and thus is not amenable to any insights from other 'quarters'. To sharpen this claim, he invented the term 'syntactocentrism' and thereby suggests that this concept of generative grammar is more of an ideology, an 'ism', than an approach that lends itself to participate in the open-minded 'fights' taking place in science.

In this book, I will explore to what extent this characterization holds in light of recent models of minimalist grammar, and I will ask what lessons can be learned from applying these recent conceptions to specific phenomena at the syntax-pragmatics interface.

In Chapter 2, in order to shed some light on the 'unstated background' Jackendoff refers to, I will provide a comprehensive account of syntactocentrism by outlining basic beliefs concerning the mental architecture of the language faculty assumed within this approach. In doing so, I will present the general idea of this model and also clarify why many scholars, including Jackendoff, regard the early version of this conception as a reasonable view due to the historical context it emerged in. I will compare early conceptions within this architectural framework with most recent versions of minimalist syntax and examine how mapping to the interface components is accomplished in Merge-based systems. Thereby, I will highlight current approaches to phonology and semantics that, to my mind, provide a good starting point for investigating minimalist mapping procedures.

Chapter 3 deals with the account of encoding different levels of meaning of a sentence by enriching the syntactic apparatus. More specifically, I will focus on syntactocentric accounts of the syntax-pragmatics interface, and I discuss analyses of one specific phenomenon, namely the pragmatics of left-periphery-movement in German. I will point out several problems with these cartographic analyses and present empirical phenomena that challenge these approaches.

Chapter 4 presents minimalist approaches to the syntax-pragmatics interface and focuses on the recent shift from 'representational' to 'derivational' syntactocentrism. After outlining this architectural change in the model of grammar, I will discuss minimalist accounts of the phenomena discussed in Chapter 3. I will argue for an approach where the domain of syntactic operations (here: displacement) is shaped by operations belonging to the cognitive component of pragmatic competence.

In Chapter 5, I will turn to prominent theoretical alternatives to syntactocentrism and first concentrate on some basic ideas within the general movement of Cognitive Linguistics. Having illustrated this view, which contradicts generative grammar in many ways, I will look at Jackendoff's own approach, the 'Parallel Architecture', which can be regarded as an intermediate position between the two extremes of Cognitive Linguistics and syntactocentrism. In the final section of this chapter, I will reflect on the question whether there is any perspective of convergence between syntactocentrism and its theoretical alternatives.

In Chapter 6, I conclude by summarizing the main results of this book and by turning to the question whether these results vindicate the notion of syntactocentrism as introduced by Jackendoff.

Overall, this book explores the conjecture that many of the objections to syntactocentrism, as they are formulated in alternative frameworks, disappear once the consequences of recent derivational syntactocentrism are taken seriously. To investigate this hypothesis, I reduce the comparison of recent syntactocentrism and other approaches to tractable size and focus on case studies at the syntax-pragmatics interface. Before I start with grounding the notion of syntactocentrism, let me add a cautionary note. This book rests on the assumption that both syntactocentrism and its theoretical alternatives – be it the Parallel Architecture or Cognitive Linguistics – belong to one single paradigm within linguistics that investigates language as a mental entity. Since the issues addressed in this book inherently require an extensive discussion of concepts assumed within mainstream generative grammar, I ask those readers that are more committed to the non-generative alternatives to practice the open-mindedness mentioned at the outset of this introduction and to adopt the view that there is no litmus test to determine membership in the category of mentalist

linguistics. Rather, as with other categories, the different approaches, even if disagreeing in various respects, are connected by family resemblance – a concept well known in some branches of non-generative linguistics. Accordingly, looking into generative linguistics may promote an understanding of the category as a whole, even if only in the sense of sharpening one's own account.

CHAPTER 2

# Syntactocentrism

## Foundations and minimalist implementation

Jackendoff has repeatedly pointed out that the "assumption of 'syntactocentrism' [...] was never explicitly grounded" (Jackendoff 2003: 655), that is, according to him, this concept has been introduced without serious argument. And indeed, when Chomsky launched this special perspective on the mental architecture of the human language faculty in the late 1950s, it was a new approach and thus, as a matter of fact, explicitly marked as a tentative assumption. However, some crucial concepts underlying the syntactocentric view were anything but new. In order to approach the notion of syntactocentrism, let me first clarify the general framework this conception is situated in.

What was not entirely new and what critics such as Jackendoff have also subscribed to over the years is the mentalist perspective on language resting on a "'capital of ideas' accumulated in the premodern period" (Chomsky 1966: 3). In his attempt to trace back the historical roots of this perspective, Chomsky especially refers to Descartes, who denied that the soul of animals were of the same kind as ours. According to Descartes, this crucial difference between man and animal manifests itself most clearly in the fact that an animal "never [...] arranges its speech in various ways [...] in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do" (Descartes 2003 [1637]: 38). Like Descartes in his reflection on human uniqueness, Chomsky places a premium on this capacity, to which he refers as the "'creative aspect' of ordinary language use" (Chomsky 1966: 4–5). To explore this aspect, adopting the mentalist view of Descartes, Chomsky assumes that a language user must have a mental capacity that enables this 'creative aspect' of language use. So, in contrast to the actual use of language in concrete situations, dubbed 'performance', this mental capacity was referred to as 'competence', as "the speaker-hearer's knowledge of his language" (Chomsky 1965: 4). Whatever the precise nature and format of the distinction between competence and performance (for recent discussion, cf. Trotzke, Bader & Frazier 2013), many

approaches since the 'cognitive revolution', which was inaugurated in the late 1950s, are committed to the view "that it is essential to consider language as a cognitive (mental) system" (Goldberg 2006: 4). However, controversies emerge with the exact formulation of this 'grammatical knowledge'. And here is where the notion of syntactocentrism comes into play.

Chomsky conceived of the speaker's knowledge as containing both a finite set of symbols, out of which sentences can be constructed, and a finite amount of combinatorial operations, a "system of rules that we can call the *grammar* of his language" (Chomsky 1964: 9, emphasis in the original). Assuming this general view of grammar, Chomsky formulates the syntactocentric claim that "a grammar contains a syntactic component, a semantic component, and a phonological component. The latter two are purely interpretive" (Chomsky 1965: 141).

In this chapter, I will provide a comprehensive characterization of the basic assumptions underlying this architecture. Given this conceptual background, I will outline current minimalist approaches to the syntactic, the phonological, and the semantic components. Following Chomsky's (1965) threefold definition of grammar given above, after introducing basic aspects of the syntactic component in Section 2.1, I will turn to the phonological component and show in what sense it is regarded as 'purely interpretive', both in the past and today. Here, I will introduce the notion of cyclicity, which will become essential for later discussion in this book. Finally, in Section 2.3, I will sketch the generative conception of the semantic component by first building on the notion of cyclicity and then illustrating an approach that reduces the operations at the syntax-semantics interface to a minimum.

As we will see in this chapter, the main motivation for assuming a syntactocentric conception of grammar is to describe a single system of generating rules that can both account for the ability to produce (phonetically) and to understand (semantically) an infinite range of sentences (for recent discussion of this motivation and relevant contributions, cf. Sauerland & Trotzke 2011). In the context of matching this rule system to the special needs of producing and understanding expressions, I will illustrate that both the phonological and the semantic component need some inventory of interpretive rules added to operations that generate the syntactic structure. Since Chomsky's (1965) seminal formulation of syntactocentrism, many approaches both to the phonological and to the semantic component have been developed. I cannot do justice to them here. My aim is merely to highlight two characteristics of the architecture that were always crucial components of this model: the necessity of a cyclic organization of grammar and the attempt to reduce the syntactic machinery to a minimum.

## 2.1 The syntactic component

In this section, I will illustrate two major goals of modeling the syntactic component: descriptive and explanatory adequacy. In Section 2.1.1, I will point out the argument that in order to achieve descriptive adequacy, that is, in order to account for all observed arrangements of data, the grammar must be of a certain complexity. In Section 2.1.2, I will focus on later developments and then show how the structure-building operation Merge incorporates major insights of earlier versions of the theory, and how Merge provides a minimal, but nevertheless sufficient approach, thereby meeting criteria that go beyond descriptive adequacy (and, in fact, even beyond explanatory adequacy, cf. Chapter 4).

### 2.1.1 The formal complexity of natural language

In the early days of generative grammar, the focus was on the development of rule systems that possess the appropriate computational properties to account for the 'creative aspect' of language use. These rule systems had to be finite, since the mental resources of humans are limited, but they should still capture the faculty of producing and understanding an indefinite number of sentences in an indefinite range of new situations, and thus, they should provide a formal account of the human capacity to "make infinite employment of finite means" (Humboldt 1999 [1836]: 91). To formulate such rule systems, Chomsky followed Bar-Hillel (1953: 165), who argues for "evaluation of [...] recent investigations on recursive definitions" within empirical sciences such as linguistics. Accordingly, Chomsky (1957) discusses different finite rule systems that operate with recursive procedures, that is, with loop-like devices that allow rules to apply to their own output (for an overview of Chomsky's early discussion of adequate rule systems, see Lasnik 2000: 12–23).

The first computational device discussed by Chomsky is a finite-state machine. To understand this type of grammar, consider, for example, how this device accounts for the following sentence (cf. Chomsky 1957: 19–20):

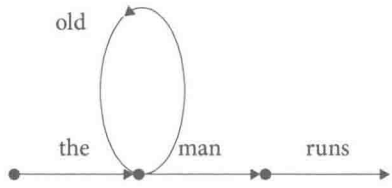
(1) The man runs.

Using a finite-state machine to model the generation of sentences like (1), we can represent the grammar graphically in the form of a so-called 'state diagram'. In such a diagram, or 'graph', the generation of a structure proceeds from an initial state to a final state in the direction of the arrows, where "[t]he 'states' are the junction points in the graph and the [...] letters produced for a transition are given beside the corresponding line" (Shannon & Weaver 1949: 15):

(2) the man runs

In order to generate an infinite number of sentences, Chomsky extends this grammar by adding closed loops, as shown in (3):

(3)




Due to this loop-like device that gives rise to recursion, the grammar can generate an infinite number of expressions like (4):

(4) The (old, old, old...) man runs.

However, Chomsky (1957:21–25) argues that finite-state grammars are formally incapable of modeling natural languages like English because such rule systems cannot account for so-called non-local dependencies. To put it more technically, Chomsky asked the question where the grammar of natural language falls on the complexity hierarchy of formal languages. He showed that a particular type of recursion is essential to drawing the line between the phrase structure models of language proposed in his work and models of language prevalent in contemporary structuralist thinking. In particular, Chomsky (1956, 1959) showed that self-embedding involves the kind of complexity that requires (at least) context-free grammars, rather than less complex types of grammar (specifically, as we saw above, finite-state devices). Chomsky (1959: 148) defined this notion of self-embedding as follows (*I* is the identity element, i.e. zero, and $\Rightarrow$ indicates a derivation involving rewrite operations):

(5) A language L is *self-embedding* if it contains an A such that for some $\varphi$, $\psi$ ($\varphi \neq I \neq \psi$), $A \Rightarrow \varphi A \psi$.

The definition characterizes as self-embedding any language that contains a string A and allows the derivation from A of a string that properly contains A, that is, A is preceded and followed by two non-trivial strings. Chomsky (1957) went on to show that patterns such as (6) exist in English. These patterns satisfy the definition of self-embedding in (5):

(6) $[\textit{The man}]_a\ [\textit{the dog}]_a\ [\textit{bit}]_b\ [\textit{runs}]_b$.

In these cases, the subject of a sentence (indicated with 'a') and the verb (marked with 'b'), though morpho-syntactically connected by agreement, can be far away from each other. In addition to examples in which two *a*s follow two *b*s, consider now, again, the fact that natural languages allow for infinite embedding. That is,

we can extend our structure (6) further by adding another sentence inside the sentence *the dog bit*, as shown in (7):

(7) $[\textit{The man}]_a\ [\textit{the dog}]_a\ [\textit{the girl}]_a\ [\textit{loves}]_b\ [\textit{bit}]_b\ [\textit{runs}]_b$.

As should be clear from the illustration so far, center-embedded structures like (6) and (7) show the general property of "*n* occurrences of *a* followed by *n* occurrences of *b* and only these" (Chomsky 1957: 21). Crucially, a finite-state grammar cannot correspond to that property because it computes a sequence strictly local, that is, it only 'knows' what state it is currently in and what to do next. Consequently, it does not 'know' what states it has been in, let alone how many times it has been in some particular state. In other words, the finite-state grammar is ignorant of the number of *a*s and *b*s it has already generated, and, therefore, it cannot ensure an equal number of *a*s and *b*s, and it cannot account for the end product of such derivations: a string with a mirror image pattern ($a_1 \ldots a_2 \ldots a_n \ldots b_n \ldots b_2 \ldots b_1$), see Chomsky (1956) for a more rigorous presentation of the argument.

To account for cases that cannot be described in terms of a finite-state grammar, Chomsky (1957: 26–33) discusses a rewrite-rule system. Adopting Chomsky's notation, we can formulate the following rules to capture our example (1):

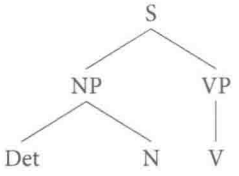
(8) S → NP + VP (S = Sentence; NP = Noun Phrase; VP = Verb Phrase)
NP → Det + N (Det = Determiner)
VP → V
Det → the
N → man
V → runs

The rules given in (8) consist of one symbol on the left side, followed by an arrow (standing for 'rewrite as'), followed by at least one symbol. When we apply these rules to generate our sentence (1), the derivation of the sentence proceeds from step (i) to step (vi), as (9) spells out in detail. In particular, every step of the derivation consists of rewriting one symbol by another (sequence of) symbol(s).

(9) S
NP + VP (i)
Det + N + VP (ii)
Det + N + V (iii)
the + N + V (iv)
the + man + V (v)
the + man + runs (vi)

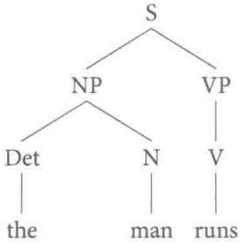
Accordingly, the syntactic component contains rewrite rules like (8) in order to generate phrase markers such as (10):

(10)




Then, lexical items are inserted into the terminal positions of the phrase marker, as the rules in (9) dictate, resulting in the tree given in (11):

(11)




Now, recall the more complex structures in (6) and (7). In order to account for these cases within a system of rewrite rules, we only have to replace the rule for NPs by a rule that reintroduces S, as shown in (12):

(12) NP → Det + N + S

By reintroducing an abstract symbol like 'S' into the derivation, the rewrite-rule system, like the finite-state grammar sketched above, contains a loop-like device and thus allows for recursion. Specifically, according to (12), the NP *the man*, containing a determiner and a noun, can be extended by a whole sentence (*the dog bit*). As this sentence contains its own NP (*the dog*), the rule given in (12) can apply again and thus extend the NP *the dog* by adding another sentence (*the girl loves*). Crucially now, in contrast to the loop-like device of a finite-state machine mentioned above, the rewrite rules capture the long-distance dependencies of sentences like (6) and (7) by ensuring that the number of *a*s corresponds to the number of *b*s. The essential difference lies in the use of non-terminal symbols like 'S'. A symbol like 'S' introduces *a*s like *the dog* and *b*s like *bit* into the derivation simultaneously. Accordingly, unlike finite-state grammars, the rewrite-rule system, due to this simultaneous insertion, makes sure that it generates an equal number of *a*s and *b*s, that is, in our case, an equal number of NPs and VPs.

Let us stop here. Of course, as noted by Chomsky (1957) himself, the rule system presented so far still cannot account for significant structural aspects