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.

Download date: 09 Jan 2019
6. Conclusion

The model of syntax developed in Chomsky’s (1995) Minimalist Program has a limited empirical scope because it relies on feature strength, which can be held responsible for some, but not all aspects of syntactic parametrization. This has led us to argue that feature strength should be replaced by constraint interaction. Constraint interaction allows us to deal with a wider range of syntactic phenomena. These include the prime topic of investigation in this book: the syntactic behavior of subjects and complementizers.

The Derivations & Evaluations model of syntax, originally proposed by Broekhuis & Dekkers (to appear) and further developed in this book, consists of two components: a generator and an evaluator. The generator is identified with Chomsky’s (1995) computational system (CHL), which contains the operations Select, Merge, and Move, and acknowledges feature checking as a motivation for movement. The evaluator functions like in any OT syntax: it evaluates syntactic structures delivered by the generator with respect to constraint rankings. The evaluator can only function if the generator overgenerates. Since Chomsky’s CHL is not originally designed to do so, we have modified it in three respects. Each of these modifications has proven to be fruitful in the analysis of subjects and complementizers.

First and foremost, syntactic structure is freely available in CHL. By incorporating Grimshaw’s Extended X-bar Theory into D&E, syntactic phenomena can be related to clause size. Throughout this book, we have argued that whenever subject movement is marked in comparison to object movement, this is caused by the absence of CP. Other phenomena that can be analyzed along these lines are deletion patterns in English prepositional relative clauses, complementizer placement in clausal complements, and in particular French Stylistic Inversion.

Second, CHL disposes of the operation Delete. This operation deletes elements without semantic content in an arbitrary fashion so that the evaluator can choose between a range of possible pronunciation patterns. Thus, the distribution of pronounced and empty relative pronouns and complementizers within and across languages can be reduced to the interaction of a small number of violable constraints.

And third, movement need not be feature-driven. The operation Move applies freely in CHL as long as subjacency is respected. Excessive movement is filtered out by the economy constraint STAY during the evaluation. Examples of non-feature-driven movement include: intermediate steps to prevent subjacency violations, short movement of deleted relative pronouns, and movement of non-subjects to SpecIP in French declarative clauses.

The division of labor between the generator and the evaluator is partially an empirical matter. We have assumed that recoverability and subjacency are operative in CHL. However, as soon as these principles turn out not to be surface-true in individual languages, they should be reformulated as violable constraints. Similarly, syntactic principles are only allowed to be members of the set of violable constraints if there are contexts in which they are not surface-true. Constraints
which are always surface-true should be reformulated as properties of the generator. Also computational issues play a role in determining the optimal shape of the generator. It should be investigated if algorithms can be constructed which ensure that the model of syntax proposed in this book can be successfully embedded in a theory of linguistic performance. Particular attention should be paid to the evaluation of infinite candidate sets. If insurmountable problems arise, the generator should be modified in such a way that it produces finite candidate sets.

Another possible object of future inquiry is the constraint inventory. In this book, we have (often implicitly) assumed that constraints should receive simple definitions, that constraint inventories lead to typological predictions that should be borne out as much as possible, and that entailment relations between constraints should be avoided. In addition, membership of a family of constraints could be required, which means that constraints should be defined in accordance with the schema that characterizes one of a small number of well-established families of constraints. In this book, the majority of constraints belong to the economy, parse, alignment, or no-empty-structure families. A possible next step is to require that constraint families be grounded (motivated extra-syntactically), where the conflict between ease of articulation (economy constraints) and ease of perception (parse and alignment constraints, constraints banning empty structure) could play a central role.