[WLG]
WIENER LINGUISTISCHE GAZETTE

Derivations as representations: News from the computational frontier

Thomas Graf

Sonderdruck aus: *Wiener Linguistische Gazette* (WLG) 82 (2017): 61–69

Themenheft *11-11-17. Festschrift für Martin Prinzhorn*
Hg. v. Clemens Mayr und Edwin Williams

Universität Wien · Institut für Sprachwissenschaft · 2017

Eigentümer, Herausgeber und Verleger:
Universität Wien, Institut für Sprachwissenschaft
Sensengase 3a
1090 Wien
Österreich

Redaktion: Christian Bendl, Mi-Cha Flubacher (Angewandte Sprachwissenschaft),
Stefan Schumacher (Allgemeine und Historische Sprachwissenschaft)
Kontakt: wlg@univie.ac.at
Homepage: *http://wlg.univie.ac.at*

ISSN: 2224-1876
NBN: BI,078,1063

Die *Wiener Linguistische Gazette* erscheint in loser Folge im Open-Access-Format.
Alle Ausgaben ab Nr. 72 (2005) sind online verfügbar.

Derivations as representations: News from the computational frontier[*]

Thomas Graf

Stony Brook University

1. Introduction

Ever since McCawley (1968) one of the fundamental questions of linguistic theory has been whether formalisms should be construed as derivational or representational in nature. The former focuses on how structures are built in an incremental fashion from pre-defined atoms via structure-building operations, whereas the latter considers all possible structures and filters out the ill-formed ones via constraints. Even within Minimalism, proposals span the gamut from Strict Derivationalism (Stroik 2009, a.o.) all the way to the purely representational Mirror Theory (Brody 2000). Rather than adjudicating between the two, this squib presents several computational arguments in support of a more pragmatic view that I call *representational derivationalism*. Representational derivationalism recognizes that both approaches have unique advantages and synthesizes them into a unique perspective of syntax that opens up several new research venues.

2. Derivation trees as syntactic data structures

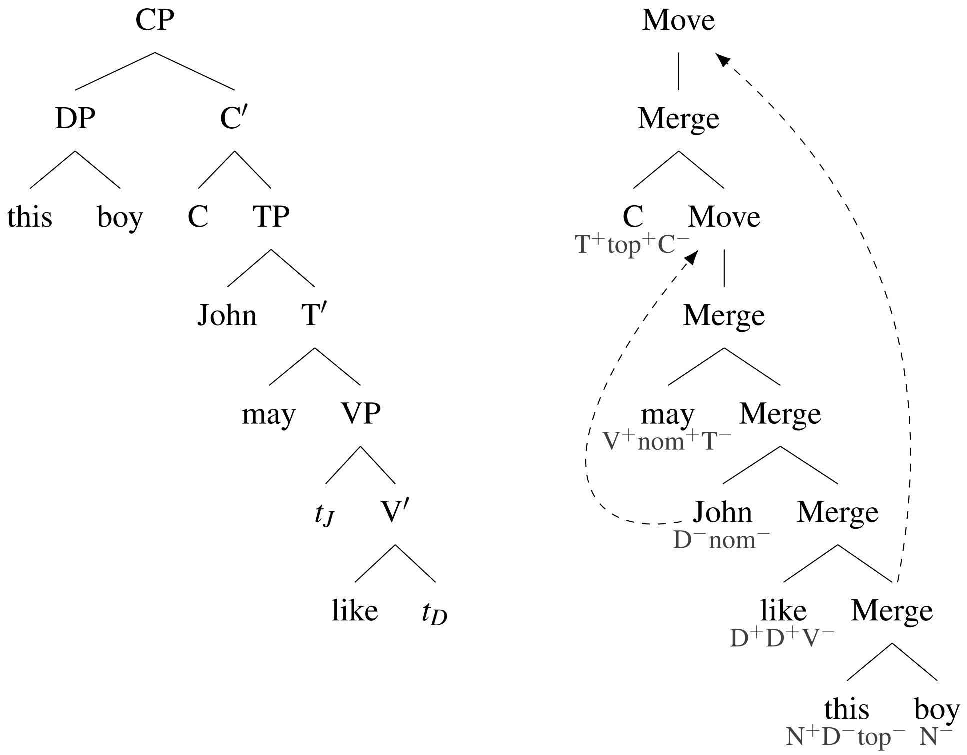
In the late 90s, computational linguists started decomposing syntactic formalisms into two base components: a finite-state tree language and a finite-state mapping from this language to the set of output structures (Mönnich 1997, 2007, 2012, Kolb et al. 2003, Morawietz 2003). While the goal was to squeeze the complicated mechanics of Tree Adjoining Grammar (TAG; Joshi 1985) and Minimalist grammars (MGs; Stabler 1997, 2011) into the well-understood confines of finite-state methods, the end result displayed a striking resemblance to Chomsky (1965)'s factorization of grammars into D[eep]-structures and transformations that convert D-structures into S-structures. Even more surprisingly, it turned out that the finite-state tree languages can be taken to encode a grammar's well-formed derivations. Rather than D[eep]-structures, then, the computational factorization builds on D[erivation]-structures.

---

[*] To Martin, whose continuous efforts to have a top-notch linguistics program in Vienna granted me the privilege of an undergraduate education that outclasses even most M.A. programs in the US.

Let us look more closely at the application to MGs (even though the general idea works just as well for TAG, GPSG, GB, and many other formalisms). In MGs, syntactic trees are built by the feature-driven operations Merge and Move. For any phrase structure tree, we can retrace the derivational steps that produced it, yielding a *derivation tree* as in (1). In the derivation tree, all interior nodes are labeled Merge or Move, and all other nodes are lexical items (movement arrows are only added for the reader's convenience here and are not part of the tree).

(1)



Note that all the information of phrase structure trees is already implicit in the derivation trees because the latter are a construction blueprint for the former. So derivation trees are a viable syntactic data structure in the sense that they do not lack vital information. This interchangeability holds even in the presence of elaborate representational constraints over phrase structure trees as those can be converted into purely derivational feature checking requirements (see Graf 2013, 2017). This addresses one half of the representationalism/derivationalism debate: a derivational approach is not impoverished, we can prove mathematically that derivation trees store just as much information as other types of structures. Moreover, derivation trees are finite-state in nature, whereas phrase structure trees are not. So from a computational perspective, derivation trees are a more economic data structure.

But a representational approach may still be preferable if it is more restricted or conceptually simpler. At least the former is provably not the case. A lengthy chain of mathematical arguments (see Graf 2013) establishes that every distinction made at the level of derivation trees can also be made at the level of phrase structure trees (or bare phrase structure sets, or any other comparable output structure). With the first possible advantage eliminated, let us focus on the question of conceptual simplicity. It is this very issue of simplicity where derivational and representational accounts can be fruitfully synthesized.

In a purely derivational version of MGs, well-formedness is determined by the feature calculus. A grammar that needs to account for even a handful of phenomena like subject-verb agreement, passive, topicalization, wh-movement and the distribution of reflexives quickly turns into an impenetrable list of lexical items with cryptic feature annotations. A single LI may be multiplied out into hundreds that differ only marginally in their features, and the size of a lexicon with a few thousand lexical items might blow up to millions. It is virtually impossible for any human to discern what well-formedness conditions such a grammar enforces.

Let us look at a concrete example of how the derivational focus on features renders simple generalizations verbose by compiling them into the feature make-up of lexical items. Consider the small MG below, where category features are indicated by $-$ and subcategorization features by $+$. The details of the feature calculus are not important here—the reader need only pay attention to how the lexicon grows with each step.

(2)
John :: D^{-}
Mary :: D^{-}
Google :: D^{-}
loves :: $\mathrm{D}^{+}\mathrm{D}^{+}\mathrm{V}^{-}$
himself :: D^{-}
herself :: D^{-}
itself :: D^{-}

This grammar builds VPs of the form *X loves Y*, where X and Y are DPs. This includes both the well-formed *John loves himself* and the ill-formed *himself loves John*. The only way to rule out the latter with the MG feature checking mechanism is to explicitly distinguish reflexives from R-expressions.

(3)
John :: D^{-}
Mary :: D^{-}
Google :: D^{-}
loves :: $\mathrm{D}^{+}\mathrm{D}^{+}\mathrm{V}^{-}$
loves :: $\mathrm{R}^{+}\mathrm{D}^{+}\mathrm{V}^{-}$
himself :: R^{-}
herself :: R^{-}
itself :: R^{-}

But that of course does not account for gender agreement between the reflexive and its subject antecedent, which necessitates additional category refinement:

(4)
John :: $\mathrm{D}^{-}_{\mathrm{m}}$
Mary :: $\mathrm{D}^{-}_{\mathrm{f}}$
Google :: $\mathrm{D}^{-}_{\mathrm{n}}$
loves :: $\mathrm{D}^{+}_{\mathrm{x}}\mathrm{D}^{+}_{\mathrm{y}}\mathrm{V}^{-}$
loves :: $\mathrm{R}^{+}_{\mathrm{m}}\mathrm{D}^{+}_{\mathrm{m}}\mathrm{V}^{-}$
loves :: $\mathrm{R}^{+}_{\mathrm{f}}\mathrm{D}^{+}_{\mathrm{f}}\mathrm{V}^{-}$
loves :: $\mathrm{R}^{+}_{\mathrm{n}}\mathrm{D}^{+}_{\mathrm{n}}\mathrm{V}^{-}$
himself :: $\mathrm{R}^{-}_{\mathrm{m}}$
herself :: $\mathrm{R}^{-}_{\mathrm{f}}$
itself :: $\mathrm{R}^{-}_{\mathrm{n}}$

Here $\mathrm{D}^{+}_{\mathrm{x}}$ and $\mathrm{D}^{+}_{\mathrm{y}}$ are placeholders for arbitrary choices of gender features. So the single entry loves :: $\mathrm{D}^{+}_{\mathrm{x}}\mathrm{D}^{+}_{\mathrm{y}}\mathrm{V}^{-}$ corresponds to $3 \times 3 = 9$ different lexical items, putting the total size of the grammar at 18 instead of the original 7. All of the 11 new lexical items are just variants of *loves* with slightly different feature values. If we were to also distinguish between singular and plural, over 40 variants of *love* would have to be listed in the grammar.

Once we consider a more realistic grammar that tries to also capture the locality requirements of reflexive licensing, the role of c-command, and the special status of exempt anaphors, the blow-up would be truly enormous. More powerful feature checking operations such as Agree can mitigate the problem of lexical blow-up, but they do not change the core problem that a purely derivational approach has to decompile well-formedness conditions into a fine-grained feature calculus that distributes the workload across a myriad of lexical items. So even though derivation trees are suitable data structures, specifying them in a purely derivational manner is cumbersome and arguably an impediment to linguistic insight.

A better approach is to treat derivation trees as representations and use constraints to restrict their shape. To ensure that constraints are still limited in their expressivity, we can choose a specific description language such as first-order logic. A derivation tree is well-formed only if it is a model of all these formulas. The need of reflexives for a gender-matching antecedent, for example, is easily expressed as a representational constraint in first-order logic.

$$\begin{aligned}&\forall x[(\text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)) \rightarrow \\ &\qquad \exists y[\text{DP}(y) \wedge \text{c-commands}(y,x) \wedge \bigwedge_{\text{gender} \in \{m,f,n\}} (\text{gender}(x) \leftrightarrow \text{gender}(y))]]\end{aligned}$$

Constraints can be added or removed without noticeable complications, whereas a derivational approach has to recompute the feature makeup of all lexical items whenever a new constraint is compiled into the grammar.

The logical view of constraints is very elegant and has been successfully used to formalize all of GB (Rogers 1998) and MGs (Graf 2013). From a computational perspective, the important thing is that the formal description language must not be more powerful than monadic second-order logic (MSO). MSO-definable constraints do not increase the power of MGs and related formalisms because they can be translated into mechanisms of the feature calculus (Graf 2013, 2017). Fortunately MSO is powerful enough to express virtually all constraints found in the syntactic literature, even transderivational ones (Graf 2010b, 2013). This shows indirectly that all syntactic constraints, even if they are stated in representational terms, can be expressed purely through feature checking. But MSO constraints are much more succinct and elegant than their feature-based equivalents, giving an edge to the representational view of derivation trees.

At this point several core insights have been established. First, the choice between representations and derivations cannot be made based on weak or even strong generative capacity. Second, derivation trees provide a computationally more parsimonious data structure than phrase structure trees or bare phrase structure sets. Third, a representational specification of well-formed derivation trees is more succinct and intuitive than a purely derivational one. In other words, representational derivationalism combines the computational advantages of derivation trees with the elegance of constraint-based systems.

But this is a purely methodological argument for representational derivationalism, which may or may not translate into concrete advantages for linguistic research. In the next section, I discuss a recently discovered parallel between phonology and syntax that could not even be formulated without representational derivationalism. This demonstrates that representational derivationalism is not just a matter of methodological beauty but an empirically fertile perspective on syntax and language as a whole.

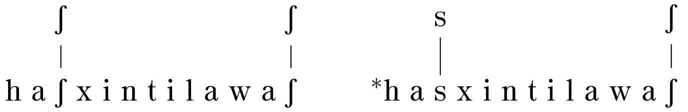
3. Parallels between phonology and syntax

I mentioned at the very beginning of this squib that if derivation trees rather than phrase structure trees are the basic data structure for syntax, then syntax is finite-state in nature. This is remarkable because that makes syntax computationally similar to phonology and morphology, whose dependencies are also finite-state (Johnson 1972, Koskenniemi 1983, Kaplan & Kay 1994). But the class of finite-state dependencies is large, so the fact that phonology, morphology, and syntax all fit into it may just be a curious accident rather than indicating a deep cognitive parallel between the three modules. However, recent work suggests that there is more to this: we can identify a very weak subclass of finite-state dependencies that make up the very core of phonology, morphology, and syntax.

Numerous findings from the last ten years have revealed that phonology and morphology use only a fraction of the power of finite-state dependencies (Heinz 2009, 2010, Graf 2010a, Heinz & Idsardi 2013, Chandlee 2014, Aksënova et al. 2016). Instead, the highly restricted subclass of *tier-based strictly local* dependencies (TSL) seems to provide an adequate fit for the large majority of dependencies (Heinz et al. 2011, McMullin 2016). Taking a hint from autosegmental phonology (Goldsmith 1976), TSL treats dependencies as local constraints over tiers. More precisely, a TSL grammar consists of a set of symbols that should be projected onto a tier and a set of forbidden local sequences that must not occur on said tier.

As a concrete example, consider sibilant harmony in Samala, which prevents two sibilants in a word from disagreeing in anteriority. Hence we find [haʃxintilawaʃ] but not [hasxintilawaʃ] or [haʃxintilawas]. A TSL grammar can capture this behavior by projecting all sibilants and forbidding adjacent instances of [s] and [ʃ] on this tier.

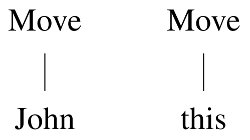
(5)



TSL thus reduces the non-local dependency of sibilant harmony to a local well-formedness constraint. This kind of “hidden locality” seems to play a central role in phonology and morphology, and it can also be found in MG derivation trees.

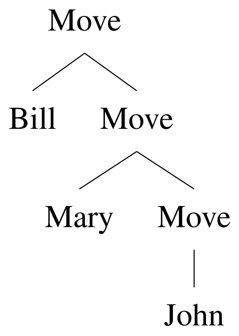
When verifying whether an MG derivation tree is well-formed, the challenging part is the long-distance nature of movement dependencies. Even with phases and successive cyclicity, there is no fixed upper bound *k* such that a mover never crosses more than *k* nodes. Object topicalization, for instance, can cross arbitrarily many VP-adjuncts. Despite appearances, movement dependencies are extremely local if one does not apply them directly over derivation trees but rather over *movement tiers*. For every movement type (wh, case, topicalization, . . .) one projects a tree tier that contains only those nodes from the derivation tree that are involved in such a movement step, either as the head of a moving phrase or as a matching Move node. For example, the derivation tree in (1) has a subject movement tier and a topicalization tier.

(6)



For a more complex example, consider the subject movement tier of the sentence *Bill thinks Mary thinks John likes this boy*.

(7)

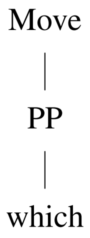


These examples show that tree tiers are trees rather than strings. They contain only those parts of a derivation tree that matter for a specific type of movement while ignoring everything else, just like the string tiers for Samala sibilant harmony ignore all material that is irrelevant for sibilant harmony.

The crucial role of movement tiers is that they make movement a maximally local relation between mothers and daughters in those tiers. The movement dependencies in a derivation are well-formed iff each projected tier obeys two constraints: every lexical item is the daughter of a Move node, and every Move node has exactly one lexical item among its daughters. Note that this makes it very easy to express certain constraints on movmeent. The adjunct island constraint amounts to projecting adjunct roots onto movement tiers as this means that the head of a phrase that moves out of an adjunct can never be the daughter of a Move node on the corresponding tier.

(8) *Sentence*: Which report did John go home without filing ⟨which report⟩?

wh-Tier:



The TSL perspective of movement is a unique view of syntax that, to the best of my knowledge, has not been explored before. Whether it provides a fully adequate picture or falls woefully short in certain respects is still an open issue, and right now it depends on several technical assumptions that are innocuous for MGs but may be problematic for Minimalism. But it nonetheless shows at an abstract level that movement, the core of Minimalist syntax, involves dependencies of comparable complexity to what we find in phonology and morphology. Without derivation trees, this result would not be obtainable because phrase structure trees have a higher degree of complexity that does not fit within TSL. But derivation trees are not enough, one also has to view them as representations that can be regulated by constraints on tiers—couching well-formedness purely in terms of feature checking precludes a TSL perspective of syntax. Hence the TSL-parallel between syntax, phonology, and morphology only surfaces when syntax is viewed through the lens of representational derivationalism.

4. Conclusion

The choice between representations and derivations is a nuanced one that a single squib cannot do full justice. Nonetheless I hope that the appeal of a Solomonic solution that fuses these two traditions into *representational derivationalism* has been aptly demonstrated. Switching from phrase structure trees to derivation trees offers many advantages, but it does not commit us to a derivational perspective of derivation trees.

References

Aksënova, Alëna, Thomas Graf, & Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 121–130. URL `https://www.aclweb.org/anthology/W/W16/W16-2019.pdf`.

Brody, Michael. 2000. Mirror theory: Syntactic representation in perfect syntax. *Linguistic Inquiry* 31:29–56. URL `https://doi.org/10.1162/002438900554280`.

Chandlee, Jane. 2014. Strictly local phonological processes. Doctoral dissertation, University of Delaware, Newark, DE. URL `http://udspace.udel.edu/handle/19716/13374`.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Goldsmith, John. 1976. Autosegmental phonology. Doctoral dissertation, MIT, Cambridge, MA.

Graf, Thomas. 2010a. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:Article 10. URL `http://repository.upenn.edu/pwpl/vol16/iss1/10`.

Graf, Thomas. 2010b. A tree transducer model of reference-set computation. *UCLA Working Papers in Linguistics* 15:1–53.

Graf, Thomas. 2013. Local and transderivational constraints in syntax and semantics. Doctoral dissertation, UCLA, Los Angeles, CA. URL `http://thomasgraf.net/doc/papers/Graf13Thesis.pdf`.

Graf, Thomas. 2017. A computational guide to the dichotomy of features and constraints. *Glossa* 2:1–36. URL `https://dx.doi.org/10.5334/gjgl.212`.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351. URL `https://doi.org/10.1017/S0952675709990145`.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661. URL `http://dx.doi.org/10.1162/LING_a_00015`.

Heinz, Jeffrey, & William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5:111–131.

Heinz, Jeffrey, Chetan Rawal, & Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58–64. URL `http://www.aclweb.org/anthology/P11-2011`.

Johnson, C. Douglas. 1972. *Formal aspects of phonological description.* The Hague: Mouton.

Joshi, Aravind. 1985. Tree-adjoining grammars: How much context sensitivity is required to provide reasonable structural descriptions? In *Natural language parsing*, ed. David Dowty, Lauri Karttunen, & Arnold Zwicky, 206–250. Cambridge: Cambridge University Press.

Kaplan, Ronald M., & Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378. URL `http://www.aclweb.org/anthology/J94-3001.pdf`.

Kolb, Hans-Peter, Jens Michaelis, Uwe Mönnich, & Frank Morawietz. 2003. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science* 293:261–289.

Koskenniemi, Kimmo. 1983. Two-level morphology: A general computational model for word-form recognition and production. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

McCawley, James D. 1968. Concerning the base component of a transformational grammar. *Foundations of Language* 4:243–269.

McMullin, Kevin. 2016. Tier-based locality in long-distance phonotactics: Learnability and typology. Doctoral dissertation, University of British Columbia, Vancouver, BC.

Mönnich, Uwe. 1997. Adjunction as substitution. In *Formal Grammar '97*, 169–178.

Mönnich, Uwe. 2007. Minimalist syntax, multiple regular tree grammars and direction preserving tree transductions. In *Model theoretic syntax at 10*, ed. James Rogers & Stephan Kepser, 83–87.

Mönnich, Uwe. 2012. A logical characterization of extended TAGs. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 37–45. Paris, France. URL `http://www.aclweb.org/anthology-new/W/W12/W12-4605`.

Morawietz, Frank. 2003. *Two-step approaches to natural language formalisms.* Berlin: Walter de Gruyter.

Rogers, James. 1998. *A descriptive approach to language-theoretic complexity*. Stanford, CA: CSLI.

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer. URL `https://doi.org/10.1007/BFb0052152`.

Stabler, Edward P. 2011. Computational perspectives on Minimalism. In *Oxford handbook of linguistic Minimalism*, ed. Cedric Boeckx, 617–643. Oxford: Oxford University Press.

Stroik, Thomas S. 2009. *Locality in Minimalist syntax*. Cambridge, MA: MIT Press.

Thomas Graf
mail@thomasgraf.net