Optimality is not a Race:
Against a Performance-Based View of Reference-Set Computation

Thomas Graf, UCLA

Problem Reference-set constraints (RCs; also known as transderivational constraints) differ from standard well-formedness conditions in that for every tree, they compute a set of output candidates called its *reference set* and pick from said set the optimal candidate(s) according to some economy metric. Well-known examples of RCs are Fewest Steps and Merge-over-Move [1], Rule I [9] and Scope Economy [3]. It has been argued in the literature ([6], among others) that if RCs have any role to play in language, it is in the parser, where they emerge as an epiphenomenon of parallel processing. The intuition is that the assembly of optimal outputs involves fewer steps, so suboptimal outputs are ungrammatical because they are discarded by the parser once the optimal candidate has been assembled. In other words, optimality is a race between candidates (Fewest Steps is one of the few RCs where this logic makes immediate sense, but for the sake of argument I will assume that it can be extended to all RCs). Given the Strong Minimalist Hypothesis, it seems indeed preferable to derive RCs from independently posited properties of the parser (see [5] and references therein) rather than treat them as a core component of narrow syntax. But I argue that the opposite is the case: If RCs have any role to play in language, the null hypothesis is for them to reside in syntax.

Argument 1 Putting RCs in the parser is not an innocent move at a methodological level. The default choice for a parsing model is the fully transparent parser, which uses only mechanisms that are already available in syntax. Any deviation from this model has to be motivated by empirically attested phenomena such as garden path sentences or local coherence [10]. Putting RCs only in the parser is such a deviation, and it isn't supported by conclusive evidence without further assumptions: RCs are supposed to distinguish between grammatical and ungrammatical forms, whereas parsing models should only account for the variable difficulty of processing structures — anything else is at odds with the competence-performance hypothesis.

Argument 2 Reference-set constraints are not race-like, nor do they involve genuine comparison. When viewed from a mathematical perspective, they turn out to be merely a different way of specifying standard well-formedness conditions (which, in the case of Minimalism, may be implemented as restrictions on the distribution of features in the lexicon). This mathematical perspective is provided by *linear bottom-up tree transducers* (lbutts; [4]), i.e. machines that take a tree as input and traverse it from the leaves towards the root while at the same time transforming it into one or several output trees. Metaphorically speaking, lbutts are to trees as SPE is to strings. The interest in lbutts stems from the fact that, when applied to a language L generated by a Minimalist grammar, they yield an output language of the same complexity as L [7]. That is to say, for every grammar G_i using only RCs that can be modelled by lbutts, there is a grammar G_j without any RCs that derives the same language as G_i. Notably, G_j preserves the RC-free part of G_i without changes. It follows that any RC R that can be modelled by an lbutt is equivalent to some constraint C that does not involve reference-set computation.

Consider Fewest Steps (FS): Given a set of convergent derivations over the same lexical items, syntax picks the derivation(s) that involve(s) the fewest instances of Move. FS is captured by the following sequence of lbutts (every lbutt can be decomposed into a sequence of lbutts, which makes them easier to define; crucially, though, this means that there are infinitely many other ways FS could have been sequenced, and the way I do it here is merely meant to aid intuition — no significance should be attributed to the details of each individual lbutt, as they are not reflected in the big lbutt directly modelling FS). The input language is the set of derivation trees of our grammar. We first have to define an lbutt R that will compute the correct reference-set, which is easily accomplished by the lbutt that may remove or add instances of Move at any node in a derivation tree (for the sake of brevity I ignore features here, although they introduce only minor complications). Now t can be rewritten as t' by R iff t' is identical to t *modulo* the Move nodes. We can restrict R such that it does not generate any trees that aren't already contained in the input language. This gives us the intended set of competitors for every choice of t. It only remains for us to implement the economy metric, to which end we define the lbutt $+Move$, which may only add Move-nodes, but not remove them. Now t is optimal iff there is no t' such that t' can be rewritten as t (i.e. there is no t' with fewer instances of Move than t that is otherwise identical to t). Some advanced mathematical theorems then tell us how to modify $+Move$ such that it does not generate any of these suboptimal candidates. So when we recombine R and the modified version of $+Move$, we get the big lbutt *FS* which will rewrite every tree in the input language by the tree(s) that was built from the same lexical items, belongs to the input language, and contains the fewest instances of Move.

Argument 3 Even if one decides to treat RCs as distinct from their corresponding well-formedness conditions (despite the main insight of argument 2), the lbutt perspective still implies that they are not race-like, because all competing candidates are generated in the same number of steps. This follows from the fact that the number of rewrite steps carried out by an lbutt depends only on the size of the input tree, not the output. Every node in the input tree has to be traversed, and whether the node has to be manipulated — and to which degree — is irrelevant for both the number of transduction steps and the overall runtime.

Argument 4 Whoever disagrees with the lbutt metric in favor of the parser-centered approach has to address the question what metric the parser uses. As noted in [5], a parallel parser which picks the shortest derivation amounts to adopting the derivational theory of complexity [8], which is incorrect under its literal interpretation [2]. Restricting this proposal to RCs is highly stipulative, so the most plausible alternative is to measure length in terms of parsing steps instead. In general, though, this gauge won't line up neatly with derivational complexity; for instance, parsers are thought to operate in a strictly local manner, such that a simple operation like topicalization may correspond to an unbounded number of slash percolation steps in the parser (cf. [5]). Without an elaborate theory of how steps in syntax and in the parser are to be put into correspondence (which has to make specific assumptions about syntax and the parser that are orthogonal to the issues covered by RCs), the claim that RCs are parsing epiphenomena is vacuous. With such a theory, on the other hand, it is by all measures a more complicated proposal than having RCs in syntax, which — as I showed in argument 2 — comes for free.

Conclusion I showed that relocating RCs to the parser is a deviation from the null hypothesis (Argument 1), whereas they naturally reside in syntax if understood as but a different way of defining standard constraints (Argument 2). Even if one rejects this dual-perspective, the lbutt model contradicts the notion that optimality is intrinsically race-like (Argument 3), which the RC-parser connection depends on. Finally, if one does not concede even this point, then the problem remains that the measure of derivation length cannot be a syntactic one for empirical reasons, whence the connection between RCs and their implementation in the parser becomes opaque (Argument 4). In sum, then, RCs reside in syntax by default; they may also be employed by other modules, but such claims need further empirical support.

References

[1] Chomsky, Noam. 1995. *The minimalist program*. Cambridge, Mass.: MIT Press.

[2] Fodor, Jerry A, Thomas G. Bever, and Merrill F. Garrett. 1974. *The psychology of language*. New York: McGraw-Hill.

[3] Fox, Danny. 2000. *Economy and semantic interpretation*. Cambridge, Mass.: MIT Press.

[4] Gécseg, Ferenc, and Magnus Steinby. 1984. *Tree automata*. Budapest: Academei Kaido.

[5] Hale, John. to appear. What a rational parser would do. *Cognitive Science* .

[6] Jacobson, Pauline. 1997. Where (if anywhere) is transderivationality located? In *The limits of syntax*, ed. Brian D. Joseph, Carl Pollard, Peter Culicover, and Louise McNally, 303–336. Burlington, MA: Academic Press.

[7] Kobele, Gregory M., Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10*, ed. James Rogers and Stephan Kepser, 71–80.

[8] Miller, George A. 1962. Some psychological studies of grammar. *American Psychologist* 17:748–762.

[9] Reinhart, Tanya. 2006. *Interface strategies: Optimal and costly computations*. Cambridge, Mass.: MIT Press.

[10] Tabor, Whitney, Bruno Galantucci, and Daniel Richardson. 2004. Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language* 50:355–370.