Compositionality and Syntactic Structure

Marcus Kracht

Department of Linguistics
UCLA
3125 Campbell Hall
405 Hilgard Avenue
Los Angeles, CA 90095–1543
`kracht@humnet.ucla.edu`

§1. The Questions

① Why does language have structure?

② What does the structure consist in?

③ Which structure does a given language have?

④ ... and how do we know?

§2. My Answers

❶ Structure exists because there is no other way to get the meanings assembled.

❷ Structure *is* the way constituents are assembled into bigger units. Structure need not be recorded (by using brackets).

❸ There may be alternative structures for sentences. We know things only within bounds.

❹ The method of inquiry is to posit a few intuitive assumptions (eg compositionality). The rest follows by straightforward reasoning.

§3. Example

What structure does the following string have?

(1) `12+7+41+3`

and how about this one:

(2) `((12+(7+41))+3)`

NB: Brackets are to be seen as actual *alphabetic symbols*.

Question: Was your answer informed by the meaning these things normally have?

§4. English

What is the structure of

(3) `Alice, Bert and Cindy sang, danced and jumped, respectively.`

and why?

§5. Definition

A language L is *weakly context free* (weakly CF) if its associated string language is CF. L is *strongly CF* if there is a compositional CF grammar for L.

> Is there a difference between the notion of weak CF language and the notion of strongly CF language? In other words: could it be that the semantics constrains the way in which syntactic functions operate? And how about natural languages?

§6. Problem Case: Dutch

Dutch shows the following dependencies:

(4) `dat` Jan$_1$ Piet$_2$ Marie$_3$ de kinderen$_4$ zag$_1$ laten$_2$ leren$_3$ zwemmen$_4$

that Jan saw Piet let Marie teach the children to swim

Huybregts (1984) has claimed that Dutch is not strongly CF *even if it is weakly CF*. But:

☆ How can we distinguish weak and strong CF languages? What is a ‘correct’ analysis?

§7. Verb Cluster Analysis

(LFG/Generative Grammar/CCG/TAG; many different derivations are conceivable, see Haider (2003).)

① With NP-cluster (GB/LFG)

(5) dat [Jan$_1$ [Piet$_2$ [Marie$_3$ [de kinderen$_4$]]]
[zag$_1$ [laten$_2$ [leren$_3$ zwemmen$_4$]]]]

② Right branching (CCG/TAG)

(6) dat [Jan$_1$ [Piet$_2$ [Marie$_3$ [de kinderen$_4$
[zag$_1$ [laten$_2$ [leren$_3$ zwemmen$_4$]]]]]]]

Generative grammar used D-structure to generate a center embedding structure.

§8. Crossed Dependencies

Subject and (complex) infinitive form a (discontinuous!) constituent.

① Local nonlinearity (Ojeda 1988): using a GPSG backbone, but allows nonlinearity of daughters.

② Calcagno (1986) uses a categorial grammar backbone, but pairs of strings as constituents (head-grammars, LCFRSs).

(7) `dat Jan Piet Marie de kinderen zag laten leren zwemmen`

§9. But ...

why not propose a center embedding?

(8) dat Jan Piet Marie de kinderen zag laten leren zwemmen

Intuition tells us that this is wrong for semantic reasons—but is there a formal proof? ☡

§10. Signs

A *sign* is a pair $\sigma = \langle e, m\rangle$, where e is the *exponent* of σ and m its *meaning*. A *language* is a set L of signs.

$$\begin{aligned}\varepsilon[L] &:= \{e : \text{there is } m{:}\langle e, m\rangle \in L\}\\ \mu[L] &:= \{m : \text{there is } e{:}\langle e, m\rangle \in L\}\end{aligned}$$

What I shall not use (but one might):

A *c-sign* is a pair $\sigma = \langle e, c, m\rangle$, where e is the *exponent* of σ, c its *category* and m its *meaning*. A *c-language* is a set L of c-signs.

§11. Grammars

① Languages are *sets* of signs.

② *grammars* are devices to generate languages.

③ *grammars* consist in certain functions that take signs as input and output a sign. For example, concatenative MERGE:

$$\mathrm{MERGE}(\langle \vec{x}, m\rangle, \langle \vec{y}, m'\rangle) := \langle \vec{x}^\frown \square^\frown \vec{y}, g(\vec{x}, \vec{y}, m, m')\rangle$$

(Notice that MERGE does not insert anything, not even boundaries! Also: g may depend on all four input parameters.)

④ a *lexicon* is a set of signs.

§12. Independence I

Let $S = E \times M$. Then for every f there are partial functions f^{ε} and f^{μ} such that

$$(9) \qquad \mathfrak{I}(f)(\sigma_0, \cdots, \sigma_{n-1}) = \langle f^{\varepsilon}(\vec{\sigma}), f^{\mu}(\vec{\sigma}) \rangle$$

G is *autonomous* if for all f, f^{ε} is independent of the meanings of the input signs, G is *compositional* if for all f, f^{μ} is independent of the exponents of the input signs. G is *independent* if it is both autonomous and compositional.

§13. Independence II

If G is independent in the strong sense then there are f^{ε}_{*} and f^{μ}_{*} such that

$$\begin{aligned}\mathcal{I}(f)(\langle e_0, m_0\rangle, \langle e_1, m_1\rangle, \cdots, e_{n-1}, m_{n-1}\rangle) \\ = \langle f^{\varepsilon}_{*}(e_0, e_1, \cdots, e_{n-1}), f^{\mu}_{*}(m_0, m_1, \cdots, m_{n-1})\rangle\end{aligned}$$

Given independence, constituent formation fails exactly if:

1. the syntactic parts e_i cannot be combined via f^{ε} or
2. the meanings m_i cannot be combined via f^{μ}.

Argumentation must separate syntactic and semantic reasons of failure. (See the recent paper by Pullum & Rawlins on the ‘X or no X’-construction.)

§14. Independence III

① MERGE is autonomous (by definition).

② MERGE is compositional iff there is a g_* such that

$$g(\vec{x}, \vec{y}, m, m') = g_*(m, m')$$

If MERGE is compositional:

$$\text{MERGE}(\langle \vec{x}, m \rangle, \langle \vec{y}, m' \rangle) := \langle \vec{x}^\frown \Box^\frown \vec{y}, g_*(m, m') \rangle$$

§15. Meanings and Expressions

I assume the following:

☞ Syntax is about expressions and only about them.

☞ Semantics is about meaning and only meaning.

☞ There is no deletion of anything in syntax.

☞ Semantic operations are restricted to identification of variables and ‘cylindrification’. Other meanings are lexical.

§16. Consequences

- There are no indices, no structural devices (brackets) in syntax unless they exist in the surface string. What is not seen has never been there! Categorial labels are abstract. AGR, NEG, C(OMP) etc are mnemonic at best! (This excludes many brands of generative grammar.)
- Meanings are 'alphabetically innocent' (Kit Fine). Names of unbound variables must be immaterial up to renaming. (This excludes most popular versions of DRT.)
- Types exist only up to ontological difference; type raising and other operations are not for free. (This excludes most brands of Categorial Grammar.)

§17. Why Is Dutch Not CF?

It is reasonable to suppose that the Dutch crossed dependencies satisfy the following.

Theorem 1 *Suppose that* $L \subseteq E \times R$ *is such that if* $\langle e, m\rangle, \langle e, m'\rangle \in L$ *then* $m = m'$. *If* L *is weakly CF then it is also strongly CF.*

Proof. By assumption, there are CF functions f_*^ε which generate the set $\varepsilon[L]$. There is a bijection $\pi : \varepsilon[L] \to \mu[L]$. Now put

(10) $$f_*^\mu(m_0, \cdots, m_{n-1}) := \pi(f_*^\varepsilon(\pi^{-1}(m_0), \cdots, \pi^{-1}(m_{n-1})))$$

This grammar is compositional, CF, and generates L. QED

So why is Dutch nevertheless not weakly CF?

§18. Alphabetical Innocence

Basic signs:

$$\langle \texttt{Jan}, x_0 = \mathsf{j} \rangle$$

$$\langle \texttt{de kinderen}, x_0 = \mathsf{c} \rangle$$

$$\langle \texttt{zwemmen}, \mathsf{swim}(e_0) \wedge \mathsf{act}(e_0) = x_0 \rangle$$

$$\langle \texttt{laten}, \mathsf{let}(e_0) \wedge \mathsf{act}(e_0) = x_0 \wedge \mathsf{thm}(e_0) = e_1 \wedge \mathsf{ben}(e_0) = x_1 \rangle$$

However, any (injective) renaming of the variables is equally 'the' meaning, eg $\langle \texttt{zwemmen}, \mathsf{swim}(e_7) \wedge \mathsf{act}(e_7) = x_{19} \rangle$.

§19. Computing Meanings

Using ‘Zeevat Merge’ (= Plain conjunction):

(11)
$$\begin{aligned}&\text{MERGE}(\langle \texttt{de kinderen}, x_0 = \mathsf{c}\rangle,\\ &\qquad \langle \texttt{zwemmen}, \mathsf{swim}(e_0) \wedge \mathsf{act}(e_0) = x_0\rangle)\\ =&\langle \texttt{de kinderen zwemmen}, x_0 = \mathsf{c} \wedge \mathsf{swim}(e_0) \wedge \mathsf{act}(e_0) = x_0\rangle)\end{aligned}$$

...or...:

(12)
$$\begin{aligned}&\text{MERGE}(\langle \texttt{de kinderen}, x_0 = \mathsf{c}\rangle,\\ &\qquad \langle \texttt{zwemmen}, \mathsf{swim}(e_7) \wedge \mathsf{act}(e_7) = x_{19}\rangle)\\ =&\langle \texttt{de kinderen zwemmen}, x_0 = \mathsf{c} \wedge \mathsf{swim}(e_7) \wedge \mathsf{act}(e_7) = x_{19}\rangle)\end{aligned}$$

No renaming derives (11) from (10)!

§20. CF structure

CF analyses more or less require the following derivation:

(13) $(\texttt{dat})[_7\texttt{Jan}\ [_6[_5\texttt{Piet}\ [_4[_3\texttt{Marie}\ [_2[_1\texttt{de kinderen}$
$\texttt{zag}]_1\ \texttt{laten}]_2]_3\ \texttt{leren}]_4]_5\ \texttt{zwemmen}]_6]_7$

σ_1 :=MERGE$(\langle\texttt{de kinderen}, m_0\rangle, \langle\texttt{zag}, m_1\rangle)$

σ_2 :=MERGE$(\sigma_1, \langle\texttt{laten}, m_2\rangle)$

σ_3 :=MERGE$(\langle\texttt{Marie}, m_3\rangle, \sigma_2\rangle)$

σ_4 :=MERGE$(\sigma_3, \langle\texttt{leren}, m_4\rangle)$

σ_5 :=MERGE$(\langle\texttt{Piet}, m_5\rangle, \sigma_4)$

σ_6 :=MERGE$(\sigma_5, \langle\texttt{zwemmen}, m_6\rangle)$

σ_7 :=MERGE$(\langle\texttt{Jan}, m_7\rangle, \sigma_6)$

§21. CF Structure

One can show that no matter how one assigns structure it is impossible to correctly manage the variables! A polyadic merge does not help, clever variable management does not help either. So we have a 'theorem':

Theorem 2 *Dutch is not strongly CF.*

§22. Conclusion

① Syntactic structure is not *form*. Is is not represented, it simply *is* the derivation tree of a sentence.

② Structure must be recovered from form. Often the evidence for syntactic structure is less clear than we think (Dowty, Sternefeld). There also are many competing analyses.

③ Syntactic structure can however be motivated from purely semantic constraints.

④ The 'proof' for structure can only work if we do not conflate syntax and semantics. Syntax does not delete.

⑤ Indices aren't part of semantics ('Alphabetic innocence'; Fine vs. Fiengo and May on identity). By nondeletion they are also not part of syntax.

⑥ Assume this and Dutch is not strongly CF.