Introduction and Overview: Syntax and its interfaces

Syntax/Semantics/Prosody Workshop 2014: Volker Struckmeier, University of Cologne

0. Introduction: Whatever do we need syntactic algorithms for?

Natural human languages differ from all (known) animal communication systems in that:

- Every human language makes available to its speakers an infinite array of *sentences*.
- Since infinitely many sentences could never be learned by heart, we must assume that humans master an *algorithm* that allows them to compute sentence properties:
- Every sentence has a set of computable meanings (i.e., whatever "Bob ate the pie" *really* means – it's not the same as "Please pass me the salt").
- Every sentence has a computable set of pronunciations (i.e. you cannot pronounce "Bob ate the pie" like "Please pass me the salt", either).

In other words, competent speakers of a language *know* which phonological forms *map* to which semantic functions – and that some word sequences ("Bob ate the pass the me salt") receive no such mapping.

Note that what is described here is pieces of knowledge: We describe what linguistic distinctions speakers know – but not, how they "do it" (the mapping, that is):

- A syntactic derivation is not the same as an "actual" processing event.
- A derivation is never "carried out" in any real-time, actual situation.
- Instead, a derivation is an algorithm devised by linguists.
- A derivation represents which formal properties connect to which functional properties.

Being a mapping relation, however, means that syntax will have to relate two completely distinct types of representations:

- The form side is concerned with the question like word order, which constituents are formed, and so on. However, it is not concerned with the meaning of the structure.
- Semantic representations are, on the contrary, not concerned with the forms that express meanings, but with the meanings themselves: These are non-linear, formless concepts (e.g.sets of situations for which a sentence holds true, etc.).

Form side	Syntax	Meaning side
linear ordering properties	←this maps to this→	no linear order at all
no meaning aspects	←this maps to this→	meaning aspects

1. What syntax is, at its most minimal core

In every language we have yet encountered, sentences

- contain arbitrary numbers of elements (morphemes, "words", larger constructions, ...),
- which are often (maybe always) combined in *constituent structures*.

While this claims is sometimes contested (for some constructions in some languages), its effects are absolutely clear in most constructions: In a purely linear grammar, important generalizations simply cannot be made (cf. Appendix A).

Therefore, even the simplest conceivable grammar must invoke an operation that builds constituents. In generative syntax, this operation is called *merge* (e.g., Chomsky 1999):

- If A and B are syntactic objects of some kind (words, constituents,...), then *merge* can build a *set* containing these two elements, which is itself another syntactic object:

1. a) Merge (A, B) → {A, B} *{A,B} is a syntactic object, so...*
   b) Merge ({A, B}, C) → {{A, B}, C} ... *merge is recursive*[1]*!*

- However, merge cannot decide *which* elements can merge (any syntactic object will do):

  Merge is incredibly stupid (under this definition[2]).

- Note merge specidies no internal order of {A,B}: set membership is not a linear property!
- However, merge not only represents aspects of *structure* building – but also aspects that used to be handled by separate operations in older theories: "movement"!

  Assume that we have merged together the structure in (2a). Assume furthermore that one of the element that we have merged, simply gets merged again (2b). For all practical purposes, we obtain what a GB theory would have called an "adjunction movement":

2. a) [HP specifier [H' head complement]]
   b) [HP complement [HP specifier [H' head complement]]]

Merge thus is only one operation – that applies in two cases:

- to merge elements that do not appear elsewhere in the structure ("external merge"), and
- merge again elements which do appear elsewhere, too ("internal merge" = "move").

In older theories, movement was seen as an imperfection that had to be "motivated", as a "last resort". This is outmoded thinking:

- Internal and external merge are two applications of the same operation, merge.
- Since internal merge is free and unrestricted, so is external merge!

---

[1] Cf. Hauser et al 2002 for the 'bio-linguistic' implications of this.

[2] As always, alternative theories exist which make merge a lot smarter. Whether or not this has any empirical advantage is completely unclear at the moment. Conceptually, a smarter operation is arguably unattractive!

2. What syntax is not – and how the interfaces resolve open questions

It is important to realize that syntax today is not "the universal theory of the sentence":

- Core syntax can generate structures which can never surface as "sentences", and which are not necessarily "grammatical" in the classic sense of the word.
- The two interfacing systems, in this view, have to do actual work: They must (minimally) reject structures which are deviant as far as the interface properties are concerned:

  - Semantically ill-formed structures will be rejected by the semantic interface.
  - Phonologically ill-formed structures will be rejected by the phonological interface.

Syntax thus systematically leaves open questions that can be handled more plausibly by the phonological or semantic component of the grammar. Syntax only decides what is syntactically possible -- nothing more or less.

At its core, syntax may therefore contain only the operation merge![3]

2.1 The mapping to semantics – reading meaning off of hierarchical structure

In the old days, some semantic properties were 'in the syntax', in that they caused certain syntactic operations (so that syntactic operations applied to semantic categories):

- *Theta roles* were handled (on the syntactic 'deep structure') and licensed DP arguments:

3. $[_{VP}\ Bob_{Ag}\ [_{V'}kills_{Ag,Pat}\ John_{\ Pat}]]$ — *V licenses arguments by roles*

- Likewise, attracting heads assigned other properties (on the syntactic 'surface structure'):

4. $[John_{Pat,\ Top}\ TOP^{0}[_{VP}\ Bob_{Ag}\ [_{V'}kills_{Ag,Pat}\ t_{John}]]]$ — *topic head TOP assigns topicality feature to John*

Operations of these types have been removed from the *core syntax* (to some extent):

- Theta roles are purely semantic properties and syntax (i.e., *merge*) does not 'check' whether predicates and their arguments 'match up'.
- Theta theory is part of the semantics – but can read roles off of the hierarchical structure:

5. a) [Peter [eat [the cake]]] = *enough arguments merge to V (word order irrelevant)*
   b) *[left [eat [the cake]]] = *structure crashes – not enough arguments merged to V!*

- Similarly, the semantic mapping can read quantifier scopes off of the hierarchy:

6. a) [QP ... [...Neg [...]]] = *QP hierarchically 'higher' = scopes over Neg*
   b) $[QP_1$ ... $[...QP_2$ [...]]] = *QP_1 hierarchically 'higher' = scopes over QP_2*

[3] There are discussions whether operations that address agreement properties (and other properties that syntax is important for) must be represented in the syntax, too – or whether they can be delegated, e.g. to the phonological component of the grammar. Other operations are discussed less and less often, afaik.

2.2 The interface to the form side

In the (bad[4]) old days (principles & parameters theories of the 80es and early 90es), syntax was responsible for word order phenomena quite directly:

- Structure building (in *x-bar theory*) involved fixing directional *parameters* (such as the *head parameter*), so that phrases were *linearized* by the syntax:

7. a) $[_{VP}\ DP_{Subj}\ [_{V'}\ V\ DP_{Obj}]]$ — *is linearly ordered by 'English syntax'*
   b) $[_{VP}\ DP_{Subj}\ [_{V'}\ DP_{Obj}\ V]]$ — *is linearly ordered differently by 'German syntax'*

- *Move*-α "moved" a phrase to a target position and left behind an (unpronouncable) *trace*:

8. [complement Y $[_{XP}$ specifier $[_{X'}$ head $t_{comp}]]$ — *complement precedes Y, specifier,... for syntactic reasons (i.e., nature of t)*

In the current system, both of these stipulations are gone:

- Syntactic trees are 'like a mobile', i.e. linearly unordered:

9. $\{_{XP}$ specifier $\{_{X'}$ X complement$\}\}$ — *is not linearly ordered by the syntax, corresponds to four linear orders:*[5]

   a) specifier X complement
   b) specifier complement X
   c) X complement specifier
   d) complement X specifier

- Since internal merge is just merge, there are no traces. The merged elements (sometimes called *copies*[6]) are identical. Thus, internal merge implies no 'overt' movement:

10. {complement {Y{$_{XP}$ specifier {$_{X'}$ head complement}}}} — *can surface as:*

    a) [complement Y $[_{XP}$ specifier $[_{X'}$ head ~~complement~~]] — *or:*[7]
    b) [~~complement~~ Y $[_{XP}$ specifier $[_{X'}$ head complement]]

- Therefore, a linear sequence of words does not necessarily represent semantics clearly:

11. a) [~~complement~~ Y $[_{XP}$ specifier $[_{X'}$ head complement]] — *structures are linearly*
    b) $[_{XP}$ specifier $[_{X'}$ head complement]] — *indistinguishable*

In sum, syntactic 'ordering' and linear 'word order' have become quite distinct things. So how does linear order enter into the picture – and why do we not get word order anarchy?

[4] In hindsight, generative grammar from that period is a rather baroque and very, very *stipulative* theory.

[5] But cf. Appendix B why this is still not anarchy.

[6] However, don't think Xerox: copies are identical, not like *two* pieces paper with the *same* print on them!

[7] This assumes that the linearization of the structure is as detailed here in the language under discussion. The example uses the German/English/French-type linearization. Other possible linearizations of structure can, of course, occur in other languages.

• Some syntactically fine structures cannot be linearized – they *crash* at the form interface:

12. a) [up [left under]] **under does not precede complement*
    b) [[left under] up] **up does not precede complement*
    c) [up [under left]] **left does not precede complement*
    d) [[under left] up] **up does not precede complement*

• Some structures have a possible linearization:

13. a) [Peter v [eat$_V$ [the cake]]] *English is head- and specifier-initial. Thus:*
    b) Peter$_{spec}$ >> v$_{head}$
    v$_{head}$ >> VP
    V >> DP
    D >> NP *And these can all be satisfied in*:

    Peter >> (v) >> eat >> the >> cake

• Thus, linearization by the form mapping introduces linear order (where possible). Syntax, in this view, is not concerned with linear order.

In order to see how this works, let's consider three word order phenomena in turn:

• *wh-in-situ* vs. *wh-ex-situ* structures (2.2.1)

• 'overt' versus 'covert' quantifier 'movement' (2.2.2)

• split movements (2.2.3)

2.2.1 In-situ and ex-situ positions of moved wh-elements

In the old days, syntax carried out 'covert' (i.e. invisible) movements quite liberally ("LF raising"[8]). Conversely, elements that appeared to occur 'too high' in the structure, could be 'moved back down' (invisibly, of course: "semantic reconstruction"). The new system invokes a lot less magical thinking and delegates these distinctions to the form mapping:[9]

14. a) *English wh structures:* [$_{CP}$ Who has [$_{TP}$ John ~~has~~ [$_{VP}$ ~~John~~ seen ~~who~~]]]?
    b) *French wh structures:* [$_{CP}$ ~~Qui a~~ [$_{TP}$ Jean a [$_{VP}$ ~~Jean~~ vu qui]]]?

[8] Cf. May 1977 – abused ever since for every case where word order did not match semantic expectations.

[9] See e.g. Richard 2010: He considers the distinction to be prosodically controlled! Cf. Richards 2014 for a theory where prosody can influence word order options.

2.2.2 Quantifier ambiguity as a form (!) property

In English, sentences with multiple quantified phrases are often ambiguous:

15. „Every man loves a woman".
    Interpretation ∃∀: *There is one woman, such that every man loves that woman.*[10]
    Interpretation ∀∃: *For each man exists some woman that the man loves.*[11]

In current theories, the ambiguity arises when the form mapping obscures meanings, as in:

16. a) *Interpretation ∃∀ spells out as:* [~~∃woman~~ [$_{TP}$ ∀man [$_{V}$ ∀ ~~men~~ love ∃women]]]
    b) *Interpretation ∀∃ spells out as:* [$_{TP}$ ∀man [$_{V}$ ∀ ~~men~~ love ∃women]]

2.2.3 Movements and split movements

Sometimes, it seems like movement can even 'tear apart' phrases which 'belong together':

17. Autos habe ich große nur schwedische gesehen.
    What is the object of "sehen"? *Arguably: [große schwedische Autos]!?*

Given the right implementation, this might just be a spellout phenomenon:

18. [~~große schwedische~~ Autos] habe ich [große ~~schwedische Autos~~] nur [~~große~~ schwedische ~~Autos~~] gesehen.

Interim summary of theoretical assumptions:

• Syntax generates sets of syntactic objects by applying (internal and external) merge.

• However, these structures are not checked for semantic & phonological adequacy.

• Rather, the form and meaning system will make their own decisions about this.

It is important to recall our general approach mentioned in the beginning: We are not talking about a processual, real-time *performance* system! No derivation is 'carried' only then to be 'filtered out' by a 'later' system. Rather:

• Derivations that *converge* represent forms that can be mapped to meanings, while

• Derivations that *crash* represent structures that can receive no such mapping, and

• Derivations detail exactly for which reasons such structures cannot be mapped.

[10] In a slightly oldfashioned manner, this interpretation is often called the Marilyn Monroe-reading.

[11] Bob loves Mary; John loves Sue; Tom loves Anne...– but there may be no single woman loved by everyone.

3. Example derivations: semantic, syntactic, and prosodic word order processes
Given the theory just outlined, there is more than one way of representing the interconnection of syntactic, semantic, and prosodic word order factors:

- Some word order changes (transparently) encode semantic differences (3.1).
- Some "movements" may be caused by information structural properties (3.2).
- Sometimes, purely prosodic factors may be at stake when word orders change (3.3).

3.1. Semantically relevant word order changes

How can an element 'move' with a semantic effect?

- Suppose, merge has successfully created the following predicate-argument structure:

19. [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]]
Semantic interface reads: *agent* *patient* *relation*
Phonological interface linearizes: *this DP* >> *this DP* >> *verb*

- Merge introduces a negation (external merge):

20. nicht [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]]
Semantic interface: ¬ (*agent* *patient* *relation)*
Phonological interface: *this head* >> *this DP* >> *this DP* >> *verb*

- An application of internal merge occurs:

21. alle Patienten [nicht [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]]]

Semantic interface is handed reading where the ∀-QP outscopes negation
= "for all patients, they are not healed" (= none of the patients are healed).

- In the end, derivations arrive at structures that map almost trivially[12] onto meanings:

22. [$_{CP}$ dass [$_{TP}$ ein einzelner A. [alle Patienten [nicht [$_{vP}$ ein einzelner A. [$_{VP}$ alle P. heilt]]]]]]
subclause *subject* *∀-high scope* ¬ *agent* *patient* *relation*

Aspects of the (semantic) meaning:	*read off structure:*
- We are dealing with a healing relation,	*V*
- The people who get healed are "alle Patienten",	*VP*
- The entity which does the healing is "ein einzelner Arzt",	*vP*
- The proposition is negated,	*[Neg vP]*
- all-QP can be interpreted with scope over negation, (∀¬)	*[QP [Neg]]*
- perspective center (?[13]) of the clause is the "ein einzelner Arzt" , and	*TP*
- the clause is semantically evaluated in context of matrix sentence.	*[C TP]*

[12] According to some, sentence-level semantics can even be understood as "just syntax near the interface" – note: the interface to language-external cognitive mechanisms!

[13] It is relatively unclear what functional load can be associated with the subject movement: Theta roles and Case are taken care of by other operations of the grammar.

3.2. Syntactically driven movements

One of the standard approaches for variable word order languages are *functional heads*: These attract phrases that are marked by special (e.g. information structural) features[14]:

23. [$_{TopP}$ [alle Patienten]$_{Top}$ Top^{0} [wohl [$_{vP}$ ein einzelner Arzt [$_{VP}$ ~~alle Patienten~~ heilt]]]]

These analyses are widely used, because they can make formidable descriptions:

- We can represent syntactic movement processes precisely, and
- we can tie syntactic positions to certain non-syntactic (e.g. information structural) properties.

However, the functional head approach is currently criticized[15] for its shortcomings:

- By their definition, the movement of the attracted phrase is *obligatory* – but this is rarely borne out by the data: E.g., in the Germanic languages, *scrambling*, *object shift*, *heavy NP shift* and related phenomena have a high degree of *optionality*.
- The optionality cannot be modeled as high or low spellouts of obligatorily moved phrases: These processes would lead to semantic effects (e.g. for QPs, see above) – which, however, often fail to materialize with information structurally marked phrases.
- Conceptually, the approach offers no explanations for our questions:

"Why does the topic move to position X?"	*is only restated as:*
"Why is the stipulated topic targer at position X to begin with?"	*= same question!*

3.3. Prosodically motivated – but asemantic – word order effects[16]

In some languages (e.g., German, Dutch, the Scandinavian languages), word order changes may have no straightforward semantic or information structural explanation:

24. Q: Who did you give the money to?
A: Ich habe das Geld dem Kellner/ dem Kellner das Geld gegeben.

(Completely identical semantics – and no information structural reason for word order change, as far as can be told: both answers good in this one context!)

Scrambling, object shift, and similar phenomena therefore often seemingly fail to receive an explanation that could be defined as the functional *trigger* for the word order changes.

Can we derive a-semantic move by the unrestricted workings of internal merge? Given the structure in (a), nothing in principle predicts that the outcome of the next internal merge will have to be (b) – rather, (c) is a viable alternative in:[17]

[14] Cf. e.g., Meinunger 2000, Frey 2004 for thorough and influential analyses of German word order.

[15] C.f., e.g., Struckmeier 2014

[16] The material in this subsection is a shameless plug for my theory of a-semantic word order factors (Struckmeier 2014).

[17] Because remember, merge really *is* stupid!

25. a) [wohl [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]]]]
b) [$_{TP}$ [ein einzelner Arzt]$_{DP}$ [wohl [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]]]]

c) [$_{TP}$ [$_{vP}$ ein einzelner Arzt [$_{VP}$ alle Patienten heilt]
[wohl [$_{vP}$ ein einzelner Arzt[$_{VP}$ alle Patienten heilt]]]]

Under this analysis:

- All arguments inside the vP are available for spellout 'on the left', since vP has "moved",
- Scrambling would normally be 'parallel' (preserve the order *subject* >> *object* >> verb)
- and could be a-semantic (vP moves, while the arguments *inside* vP, technically, do not):

26. dass [$_{TP}$ [$_{vP}$ ein einzelner Arzt /ALLen Patienten ~~helfen~~] ja wohl NICHT\
[$_{vP}$ ~~ein einzelner Arzt allen Patienten~~ helfen]] kann]

- *Semantic interpretation will not have all-QP outscope negation. Interpretation: "it is not the case that a single doctor will heal all patients" (but some? ¬∀, cf. (19)!)*

4. Summary

- Syntax is a mapping algorithm that allows us to define relations between (phonetic/ morphonological) forms and (semantic/pragmatic/information structural) functions.

- However, the model is not a processing model and does not describe performance. Therefore, it is no representation of "how a speaker produces/ interprets expressions":

- The mapping relation is static. It represents distinctions that speakers know matter.

- The influence of syntax, semantics and prosody on word order can be described:

  - The syntax derives internal mergers by its own, syntactic properties,[18]
  - the semantic interface can reject structures that are semantically problematic, and
  - the mapping to form decides to spell out copies that the syntax has generated, or
  - chose to reject structures that are problematic for form-related reasons.

- In individual languages, spellout follows some language-specific rules and regulations:

  - In German, semantic scopes (binding options, etc.) are reflected by spellout (i.e.: "pronounce the copy that reflects scopal or binding properties!"). But: English differs!

  - Some spellout decisions are prosodically motivated: E.g., in (24) above, spellout can only implement the bridge contour (/RISE... FALL\) as indicated – no other order works!

- The interface to the interpretation, on the other hand, may be universal across languages, since all languages express the same thoughts concepts, etc. – even if they *formally express* them differently (i.e., at the other interface, to the form side).

[18] For which, restrictions can be defined, if empirically warranted. We have portrayed the simplest case here.

References

Chomsky, Noam 1999: "Derivation by Phase". In: *MIT Occasional Papers in Linguistics* 18, pp. 1-42.

Frey, Werner 2004: „A Medial Topic Position for German“. In: *Linguistische Berichte* 198, pp. 153-190.

Hauser, Marc, Noam Chomsky., & W. Tecumseh Fitch (2002): "The language faculty: What is it, who has it, and how did it evolve?" In: *Science* 298, pp. 1569–1579.

May, Robert 1977. *The Grammar of Quantification*. PhD thesis, MIT.

Meinunger, André 2000: *Syntactic Aspects of Topic and Comment*. Amsterdam/ PA: Benjamins.

Richards, Norvin 2010: *Uttering trees*. MIT Press.

Richards, Norvin 2014: "Contiguity Theory". Download from http://ling.auf.net/lingbuzz/002247

Struckmeier, Volker 2014: Scrambling ohne Informationsstruktur? Prosodische, semantische und syntaktische Faktoren der deutschen Wortstellung. (= *Studia Grammatica 77)*. Berlin: de Gruyter.

Appendix A: Why linears grammars are clumsy (and probably insufficent) devices

Why would we assume that *hierarchical* aspects of syntactic structure are important? The simple answer is that linear descriptions fail to account for many word order facts:

• Linear relations between two elements X and Y by definition only include:
- X precedes Y (and does not follow)
- X follows Y (and does not precede)
- X is in the same structure as Y (precedes or follows)

• But none of these linear relations can account for, e.g. the distribution of *any*-words[19]:

1. a) Peter does not see anybody. *(good sentence)*
   b) *Peter sees anybody. *(negation needed in sentence?)*
   c) *Anybody does not sleep. *(negation precedes any-word?)*
   d) *[That Peter does not see me] worries anybody. *(no linear relation left to use)*

• It seems we probably want to exclude the "subject clause" from the linear relation. (*any*-word must be preceded by negation in its own clause – subclauses do not count)

• However, note that the problem re-appears in other structures, too:

2. a) *The friend [who did not come] worried anybody. *(relative clause must not count)*
   b) *The friend [of nobody] worries anybody. *(PPs must not count)*
   c) *[Peter and nobody else] worried anybody. *(conjoined DPs must not count)*
   ...and the list goes on...

• Note also that this approach leads to no significant predictions (we are just capturing facts when and if we find them) and thus offers no *explanatory* value.

• Therefore, stipulating exception after exception from linear rules may not take us all too far and a different aproach might be better – to unify all the exceptions by structural (read: hierarchical) means.

[19] More technically: NPI licensing is defined over hierarchical relations between licenser and licensee.

Appendix B: Merge (quite substantially) restricts possible word orders

Merge does not determine the order of merged elements in the set {A, B}.
However, that does not mean that it does not contribute to word order at all:

Consider three syntactic constituents, a head H, a specifier S and a complement C:

• If merge did not exist, there would 3! = 6 ways of ordering:

1. a) H-S-C
   b H-C-S
   c) S-H-C
   d) S-C-H
   e) C-S-H
   f) C-H-S

• However, merged structures have less orders (four, for three elements), since specifiers cannot intervene between heads and complements (e/f):

2. a) [specifier [head complement]]
   b) [specifier [complement head]]
   c) [[head complement] specifier]
   d) [[complement head] specifier]
   e) *[complement specifier head]
   f) *[head specifier complement]

• If four vs. six doesn't sound like much, consider longer structures: In the unmarked system, every additional element multiplies the possible word orders by the total number of elements:

3. a) Four elements (S, H, C and Adjunct) yield 4! = 24 word orders – four times as many as three elements had.
   b) Likewise, a sentence of 25 words would get *25 times* (!) as many possible word orders as a sentence that is only one word shorter.

• Under merge, however, every additional element X leads always and "only" to the doubling of word order options:

4.

additional element precedes structures	additional element follows structures
[adjunct [S [H C]]]	[[S[H C]] adjunct]
[adjunct [S [C H]]]	[[S [C H]] adjunct]
[adjunct [[H C] S]]	[[[H C] S] adjunct]
[adjunct [[C H] S]]	[[[C H] S] adjunct]

As we see, assuming that elements merge in a binary fashion, and into hierarchical structures, is equivalent to reducing the complexity of the word order representation – by reducing the number of possible word orders quite substantially.