

Crossing branches in RRG: How to capture them in TWG

Laura Kallmeyer

TreeGraSP Workshop # 4

26.02.2020



Introduction

- On the one hand, our TWG formalization of RRG does not allow for crossing branches.
- On the other hand, the treebank trees in RRGbank and RRGparbank have crossing branches.
- We have to find a way to transform a treebank tree into a tree without crossing branches, which allows then TWG analyses.
- The transformation should be such that no information gets lost, i.e., it should be reversible.

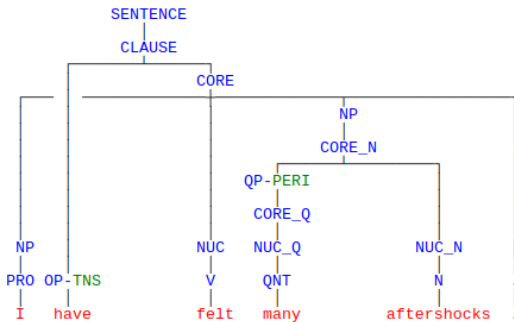
Introduction

- On the one hand, our TWG formalization of RRG does not allow for crossing branches.
- On the other hand, the treebank trees in RRGbank and RRGparbank have crossing branches.
- We have to find a way to transform a treebank tree into a tree without crossing branches, which allows then TWG analyses.
- The transformation should be such that no information gets lost, i.e., it should be reversible.

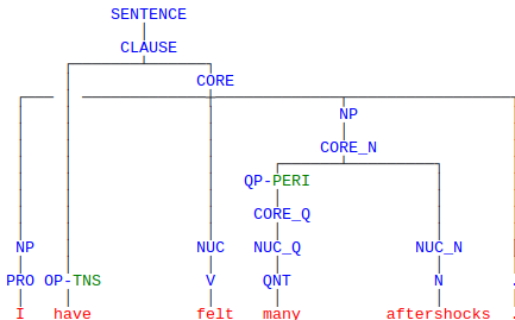
Goal of the talk: discuss different types of discontinuities in the treebank wrt. the following two questions:

1. How can we transform the tree into a tree without crossing branches?
2. How do the underlying elementary TWG trees look like?

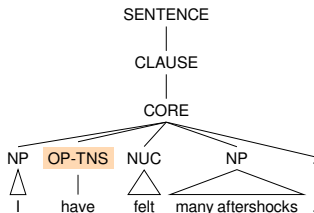
Operator projection and periphery



Operator projection and periphery

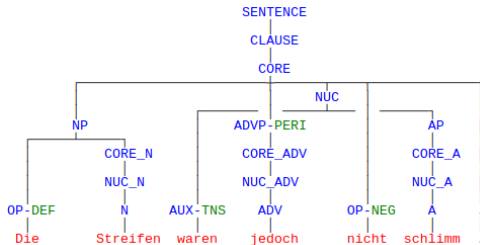


Strategy:
 Simply attach
 operator lower,
 the label tells
 us about its
 actual scope.



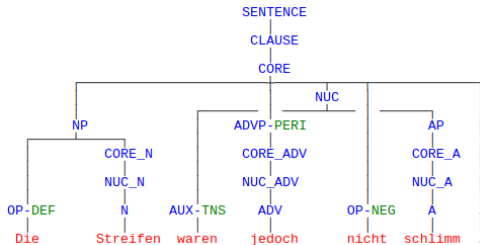
Operator projection and periphery

- (1) die Streifen waren jedoch nicht schlimm
 the (police) patrols were however not problematic
 “the patrols did not matter, however”

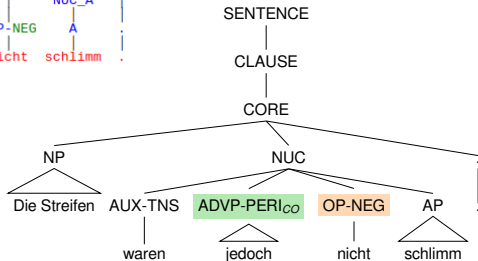


Operator projection and periphery

- (1) die Streifen waren jedoch nicht schlimm
 the (police) patrols were however not problematic
 “the patrols did not matter, however”



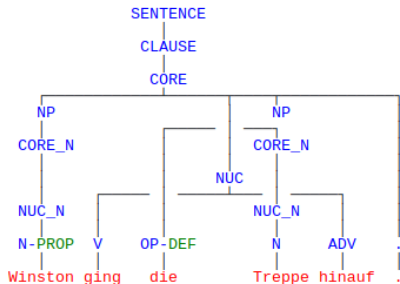
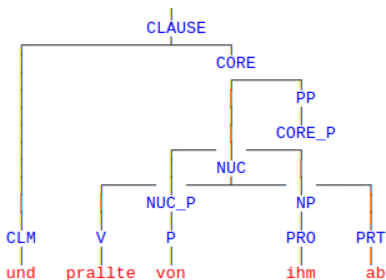
Strategy:
 Attach periphery
 element lower,
 marking its scope
 within its label.



Discontinuous NUCs

Fixed verbal MWEs (i.e., two POS tags below a NUC)

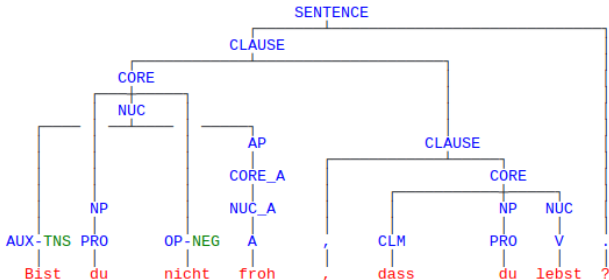
- (2) und **prallte** von ihm **ab**
 and bounced_off from him PRT
 “and bounced off (from him)”
- (3) Winston **ging** die Treppe **hinauf**
 Winston went the stairs upwards
 “Winston went up the stairs”



Discontinuous NUCs

Copula constructions

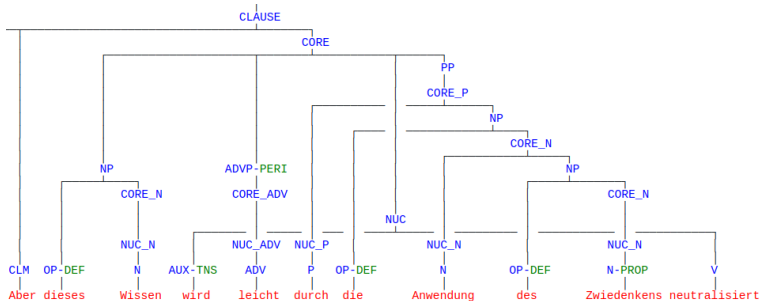
- (4) **Bist** du nicht **froh**, dass du lebst ?
 Are you not glad, that you lives ?
 “Are you not glad that you are living?”



Discontinuous NUCs

Passive

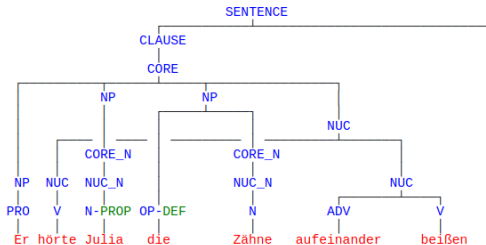
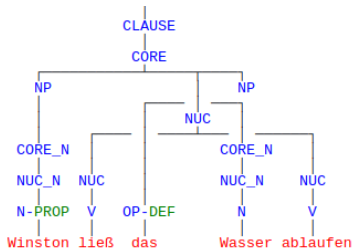
- (5) Aber dieses Wissen **wird** leicht durch die Anwendung des
 But this knowledge is easily by the application of
 Zwiedenkens **neutralisiert**
 doublethink neutralized
 “But this knowledge is easily neutralized by applying doublethink.”



Discontinuous NUCs

Complex predicates consisting of several NUCs, i.e., NUC cosubordination

- (6) Winston **ließ** das Wasser **ablaufen**
 Winston let the water drain off
 “Winston let drain off the water”
- (7) Er **hörte** Julia die Zähne **aufeinander** **beißen**.
 He heard Julia the teeth on each other bite.
 “He heard Julia snap her teeth together.”



Discontinuous NUCs

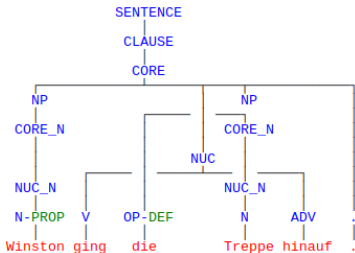
Possible strategy:

1. Introduce a feature `NUC-ID` that is supposed to be unique for each NUC node (in the original trees with crossing branches).
2. Whenever we have a NUC with more than one non-adjacent components and the material in between consists not only of OP and PERI elements: replace the NUC with two different nodes, both labeled NUC, carrying the same `NUC-ID`. Place the two components of the original NUC under the two new NUC nodes.

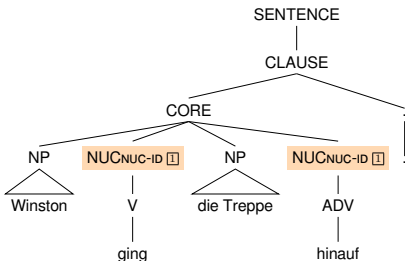
Discontinuous NUCs

Possible strategy:

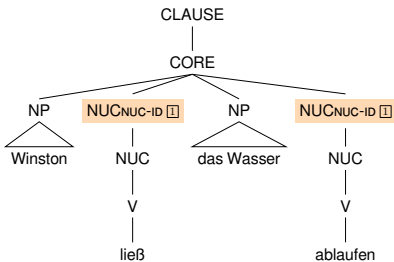
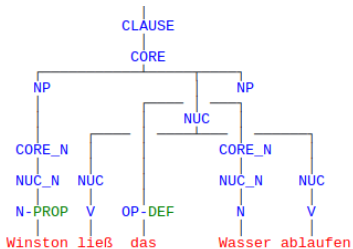
1. Introduce a feature `NUC-ID` that is supposed to be unique for each NUC node (in the original trees with crossing branches).
2. Whenever we have a NUC with more than one non-adjacent components and the material in between consists not only of OP and PERI elements: replace the NUC with two different nodes, both labeled `NUC`, carrying the same `NUC-ID`. Place the two components of the original NUC under the two new `NUC` nodes.



~>



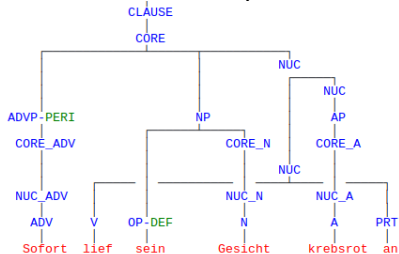
Discontinuous NUCs



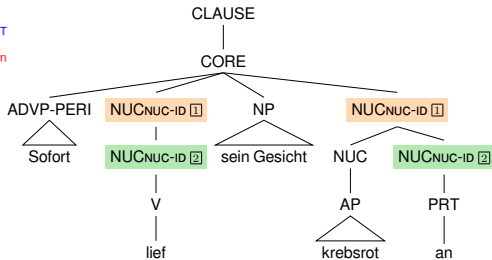
Discontinuous NUCs

A more complex example:

- (8) sofort **lief** sein Gesicht **krebsrot an**
 instantly turned his face beet-red PRT
 “his face instantly turned beet-red”

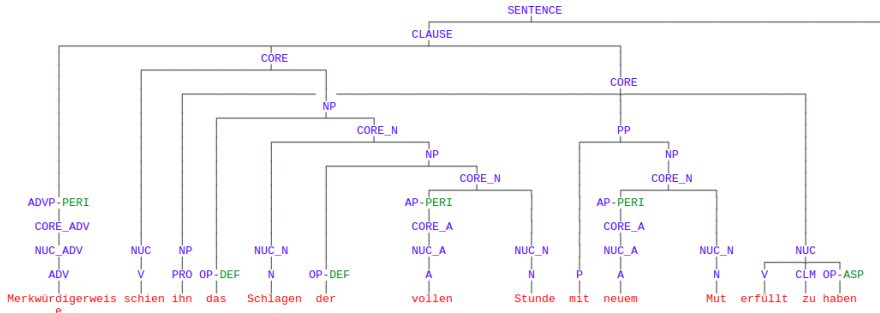


~>



Discontinuous COREs

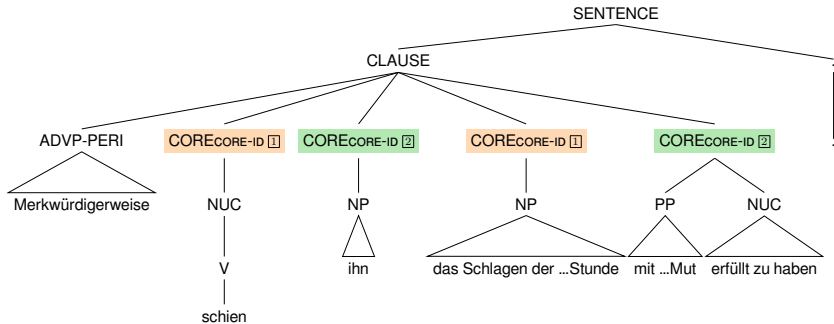
In particular in German, we have examples for discontinuous COREs.



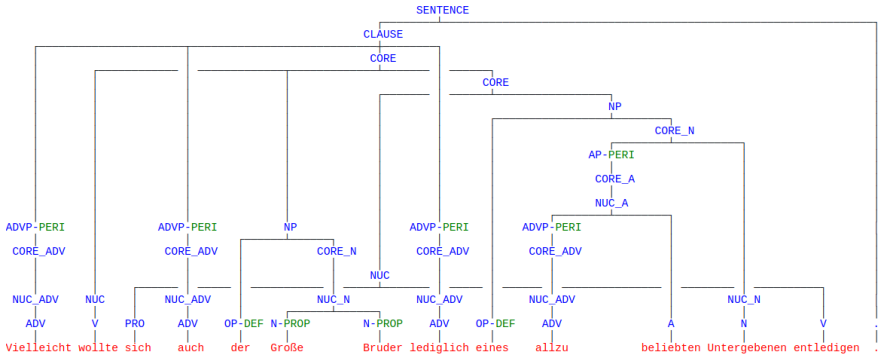
- (9) Merkwürdigerweise **schien** **ihn** **das Schlagen der vollen Stunde**
 Curiously seems him_{acc} the chiming of the full hour_{nom}
mit neuem Mut **erfüllt zu haben** .
 with new courage to have filled .
 Curiously, the chiming of the full hour seems to have filled him with new courage.

Discontinuous COREs

If we apply a similar strategy as for NUC, now with a features CORE-ID, this would lead to the following tree:



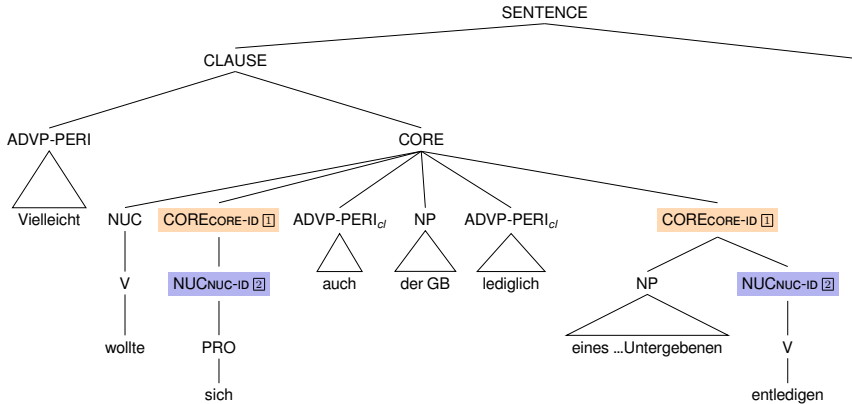
Discontinuous COREs



- (10) Vielleicht **wollte** **sich** auch **der Große Bruder** lediglich
 Perhaps wanted PRO_{refl} also Big Brother just
eines allzu beliebten Untergebenen entledigen .
 a too popular subordinate get rid of .
 Perhaps Big Brother just wanted to get rid of a too popular subordinate.

Discontinuous COREs

Transformation would lead to the following tree:

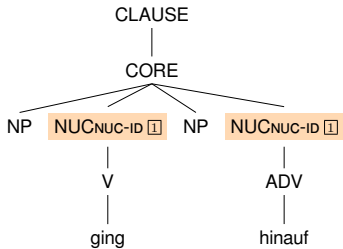


Tree Wrapping Grammar?

- We would assume that nodes with the same NUC-ID features (CORE-ID resp.) come from the same elementary tree.
- Question: How should the elementary building blocks look like for the trees we have just seen?
- Operators and periphery element are added by sister adjunction (see Tanja's talk).
- For discontinuous NUCs and COREs, the analysis depends on the specific phenomenon.

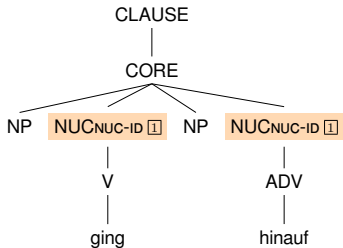
Tree Wrapping Grammar?

Fixed verbal MWEs:



Tree Wrapping Grammar?

Fixed verbal MWEs:

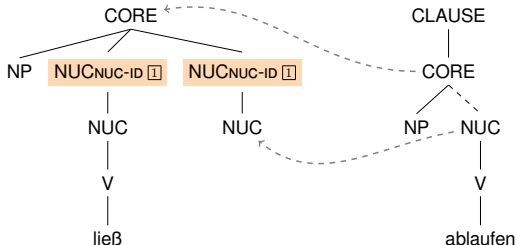


Copula and passive maybe similar?

Tree Wrapping Grammar?

Complex predicates: NUC cosubordination; merging COREs

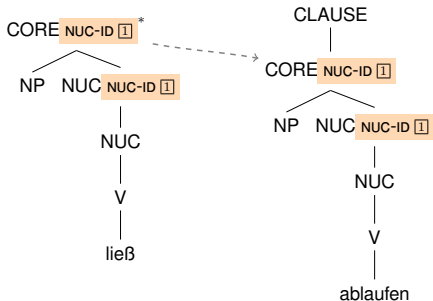
Option 1: substitution node for second NUC



Note that this would require a variant of wrapping substitution that we did not consider so far since the new NP daughter of the CORE is inserted somewhere to the left of the d-edge but not necessarily as a new leftmost daughter.

Tree Wrapping Grammar?

Option 2: Adjunction



(I like the first option better.)

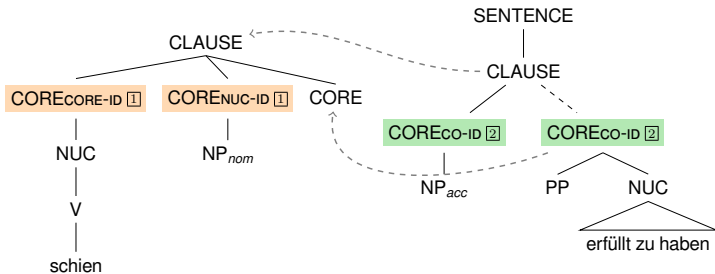
Problem with the adjunction option: NPs and Vs belonging to each other are not necessarily adjacent:

(11) **das Wasser** ließ er **ablaufen**

Tree Wrapping Grammar?

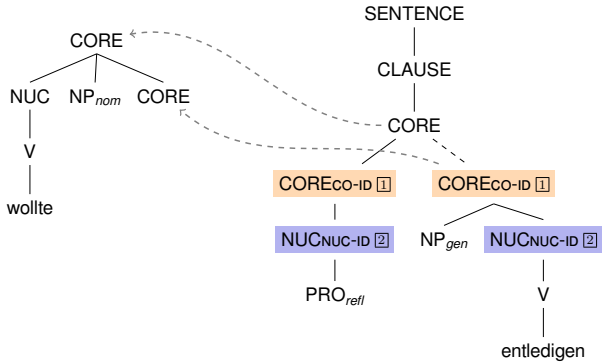
What about discontinuous CORE arguments?

CORE coordination:



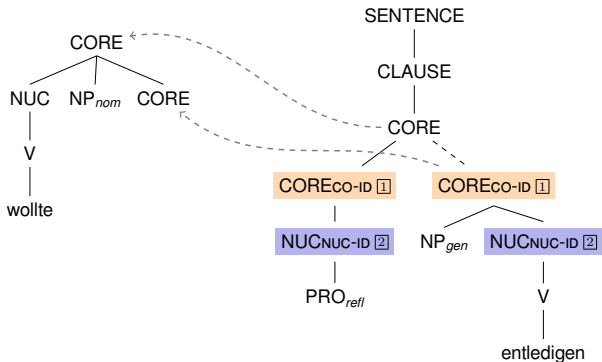
Tree Wrapping Grammar?

CORE subordination:



Tree Wrapping Grammar?

CORE subordination:



Note that in these wrapping cases, the argument contributes the higher SENTENCE and CLAUSE nodes because wrapping targets the root with the upper part of the d-edge. This is a little bit unfortunate. Maybe allow wrapping to target also internal nodes?

Tree Wrapping Grammar?

Maybe a revision of the definition of wrapping substitution?

- The lower node of the d-edge fills (as before) a substitution slot.
- The higher node merges with a node that dominates that substitution slot such that
 - ▷ no other nodes merge, and
 - ▷ all linear precedence relations are preserved.

Tree Wrapping Grammar?

Maybe a revision of the definition of wrapping substitution?

- The lower node of the d-edge fills (as before) a substitution slot.
- The higher node merges with a node that dominates that substitution slot such that
 - ▷ no other nodes merge, and
 - ▷ all linear precedence relations are preserved.

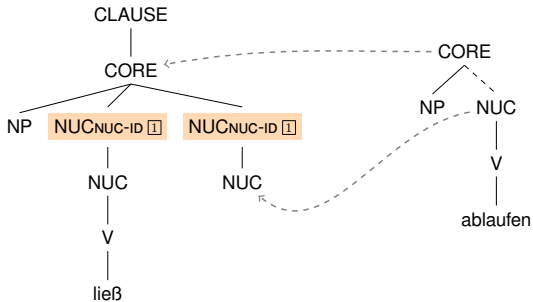
This means in particular that any daughter of the higher node becomes a (new) daughter of the merged node, and a mother of the higher node becomes a (new) mother of the merged node.

Consequently, if the higher node has a mother, we can only merge with the root node (as before). Otherwise, we can also merge with an internal node (this is new).

The newly added daughters of the higher node can permute freely with the daughters of the target node, except for their precedence relation with the d-daughter.

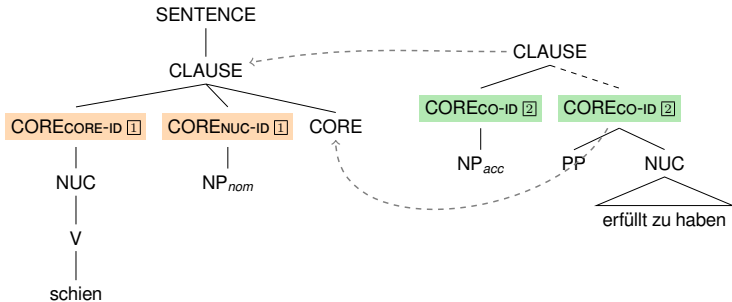
Tree Wrapping Grammar?

Revised derivation for NUC cosubordination:



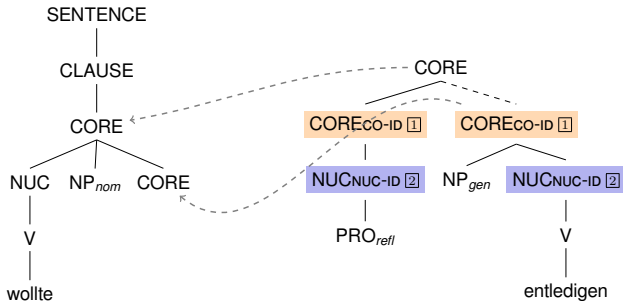
Tree Wrapping Grammar?

Revised derivation for CORE coordination:



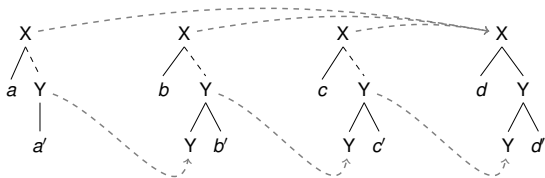
Tree Wrapping Grammar?

Revised derivation for CORE subordination:

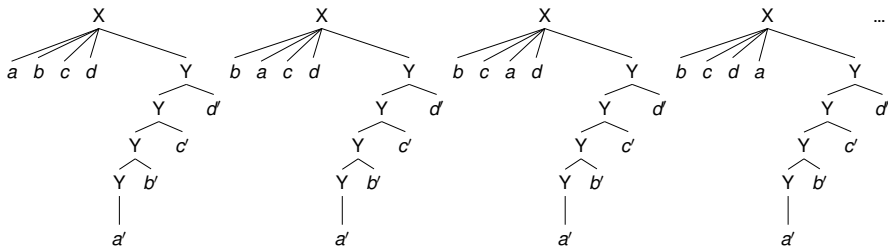


Tree Wrapping Grammar?

Note that the revised wrapping allows derivations of the following type:

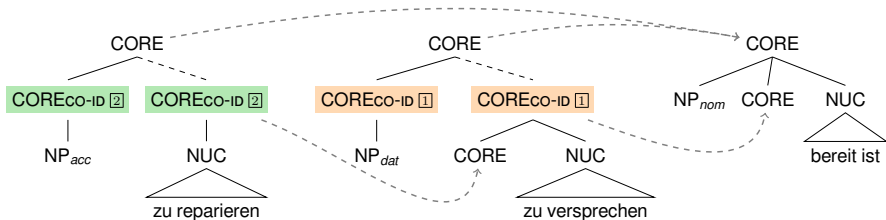


leading to all permutations of the a, b, c, d below the X node.



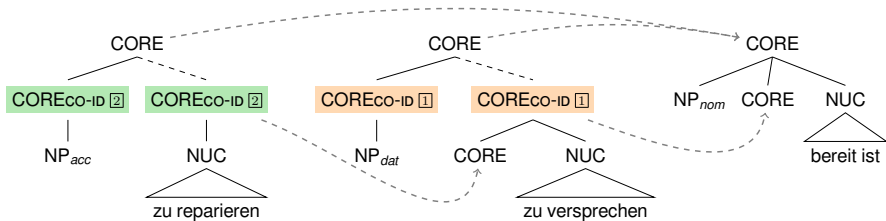
Tree Wrapping Grammar?

- (12) dass es ihm der Junge zu reparieren zu versprechen bereit ist
 that *it_{acc}* *him_{dat}* *the boy_{nom}* to repair to promise willing is
 “that the boy is willing to promise him to repair it”

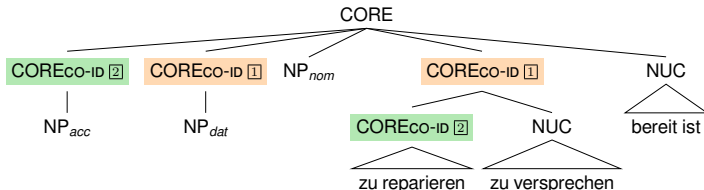


Tree Wrapping Grammar?

- (12) dass es ihm der Junge zu reparieren zu versprechen bereit ist
 that it_{acc} him_{dat} the boy_{nom} to repair to promise willing is
 “that the boy is willing to promise him to repair it”

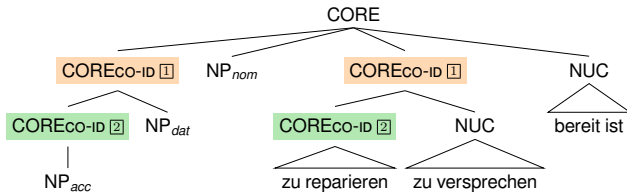


~>



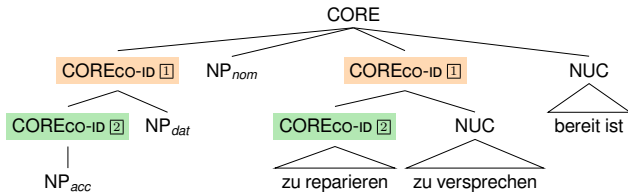
Tree Wrapping Grammar?

If we assume an RRGbank style crossing branches annotation and transform it along the lines explained above, we would actually obtain



Tree Wrapping Grammar?

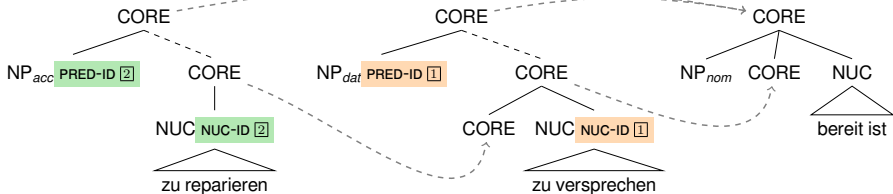
If we assume an RRGbank style crossing branches annotation and transform it along the lines explained above, we would actually obtain



Unclear, how to obtain this with wrapping the way it was just sketched. But do we want crossing branches for coherent constructions?

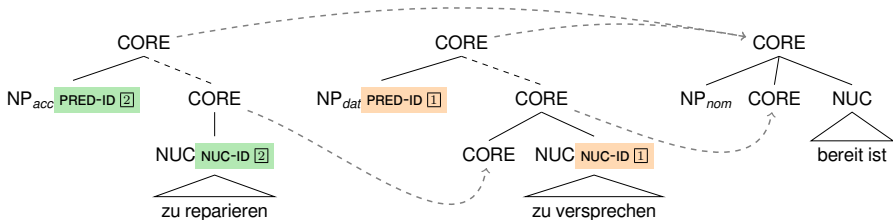
Tree Wrapping Grammar?

Alternative analysis that merges the CORE components in the mittelfeld:

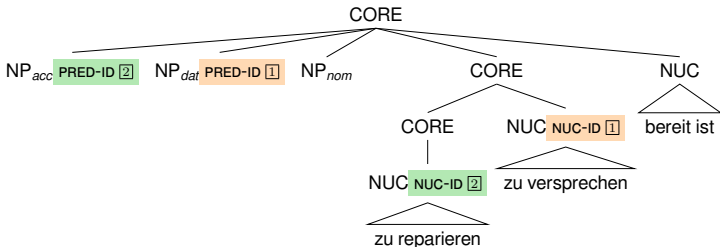


Tree Wrapping Grammar?

Alternative analysis that merges the CORE components in the mittelfeld:



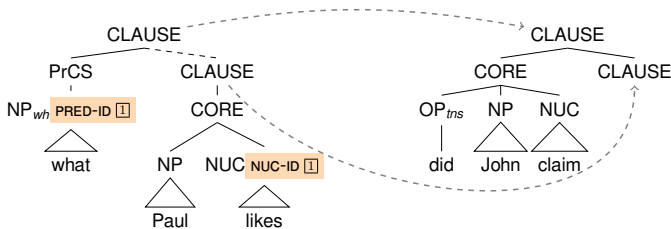
~>



Tree Wrapping Grammar?

This would be to a certain extent similar to long-distance wh-movement where RRG does not assume crossing branches:

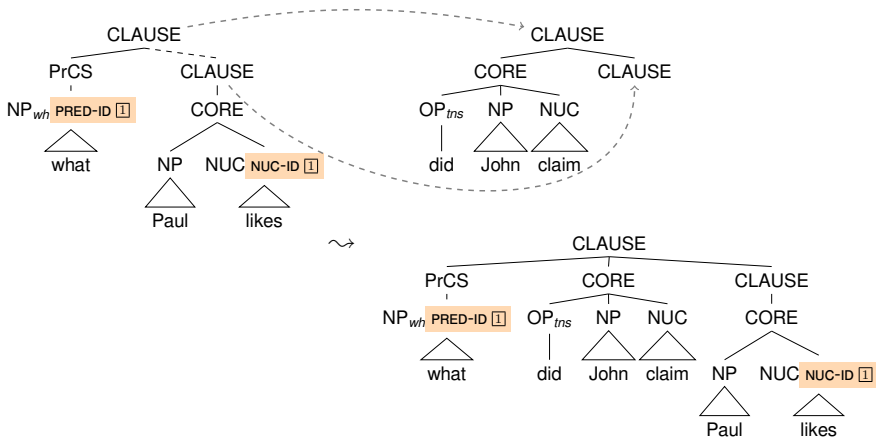
(13) what did John claim Paul likes?



Tree Wrapping Grammar?

This would be to a certain extent similar to long-distance wh-movement where RRG does not assume crossing branches:

(13) what did John claim Paul likes?



Conclusion

- “Local” cases of discontinuous constituents can be treated via different components with a shared node-id. The components are part of the same elementary tree.
- “Less local” cases of “displaced arguments” (scrambling, wh-dependencies) might not be treated as discontinuous constituents but rather via a coindexation in an elementary tree that is added by wrapping substitution.
- Extension to the wrapping substitution operation: upper target nodes are no longer restricted to roots and the daughters of the upper d-edge node and the upper target node can mix.

Appendix: Parsing with revised wrapping

Old CW (*complete wrapping*) rule:

$$\frac{(R, i, j, \Gamma_1 \oplus [(f_1, f_2, y)] \oplus \Gamma_2, ws?) \quad (D, f_1, f_2, \Gamma_3, yes)}{(D, i, j, \Gamma_1 \oplus \Gamma_3 \oplus \Gamma_2, no)} \quad \begin{array}{l} \text{root}(R) \wedge y = \ell(D) \\ \ell(\text{parent}(D)) = \ell(R) \end{array}$$

Ideas:

- Instead of requiring that we have reached the upper target node (R) of the wrapping, we could already act when being in those of its daughters that dominates the $\ell(D)$ -gap, let's call this node X .
- We then have to add all sisters of the d-daughter D and all sisters of the "gapped" daughter X . Only when this is finished, we can move to either $\text{parent}(X)$ (if all d-parents were roots) or $\text{parent}(D)$ (if this is not a root).
- For the process of combining the righthand sides of several productions, moving from left to right, we need special items that keep track of all productions.

Appendix: Parsing with revised wrapping

Modifications to the parsing items:

- We add a further list, Υ in the dotted production items. This list is empty when we start a dotted production and it should be empty when we complete it. It contains the dotted productions from wrapping trees that have to be interleaved with the production we are moving through.
- Further extension: a list of gap nodes, Ψ , with each passive X item, which have to be mapped to wrappings targeting the parent of X . This is different from Γ , it has to be “discharged” locally, i.e., at the mother node.
- We also need this Ψ with active (dotted) items, where it has to be matched to elements in the Υ list.
- The former flag $ws?$ is no longer needed.
- Item forms $(A \rightarrow \alpha \bullet \beta, i, j, \Gamma, \Upsilon, \Psi)$ or (A, i, j, Γ, Ψ) .

Appendix: Parsing with revised wrapping

New rules:

Adding a d-node:

$$\frac{(N \rightarrow \alpha \bullet M \beta, i, j, \Gamma_1, \Upsilon, \boxed{})}{(M, j, k, \Gamma_1 \oplus [(j, k, y)], \boxed{})} \quad \frac{(D, j, k, \Gamma_2, \boxed{})}{\wedge \text{leaf}(M)} \quad \ell(M) = \ell(D) = y \wedge \text{dnode}(D)$$

Predict wrapping (starts now with the leftmost sister of a d-daughter):

$$\frac{(A \rightarrow \alpha_1 \bullet \beta_1, i, j, \Gamma_1, \Upsilon, \boxed{})}{(A \rightarrow \alpha_1 \bullet \beta_1, i, k, \Gamma_1 \oplus \Gamma_2, \langle C \rightarrow M \bullet \gamma \rangle \oplus \Upsilon, \boxed{})} \quad \frac{(M, j, k, \Gamma_2, \boxed{})}{\wedge \text{dparent}(C) \wedge \neg \text{dnode}(M)} \quad C \rightarrow M \gamma \text{ a rule such that } \ell(A) = \ell(C) \wedge 1$$

Complete wrapping sister:

$$\frac{(A \rightarrow \alpha_1 \bullet \beta_1, i, j, \Gamma_1, \langle C \rightarrow \gamma \bullet M \alpha_1 D \beta_2 \rangle \oplus \Upsilon, \boxed{})}{(A \rightarrow \alpha_1 \bullet \beta_1, i, k, \Gamma_1 \oplus \Gamma_2, \langle C \rightarrow \gamma M \bullet \alpha_1 D \beta_2 \rangle \oplus \Upsilon, \boxed{})} \quad \neg \text{dnode}(M)$$

Complete d-node:²

$$\frac{(X, i, j, \Gamma_1 \oplus [(f_1, f_2, y)] \oplus \Gamma_2, \Psi_1)}{(X, i, j, \Gamma_1 \oplus \Gamma_2 \oplus \Gamma_3, \Psi_1 \oplus D \oplus \Psi_2)} \quad \frac{(D, f_1, f_2, \Gamma_3, \Psi_2)}{\wedge \text{dnode}(D) \wedge y = \ell(D)} \quad \ell(\text{parent}(X)) = \ell(\text{parent}(D))$$

¹ $\text{dparent}(N)$ = does N have a d-daughter?

² The flag $ws?$ is no longer needed because we do not extend the D item.

Appendix: Parsing with revised wrapping

Complete (does the job of the former CS (= combine sisters)):³

$$\frac{(A \rightarrow \alpha_1 \bullet X \beta_1, i, j, \Gamma_1, \Upsilon, \llbracket \rrbracket) \quad (X, j, k, \Gamma_2, \Psi)}{(A \rightarrow \alpha_1 X \bullet \beta_1, i, k, \Gamma_1 \oplus \Gamma_2, \Upsilon, \Psi)} \neg dparent(A)$$

Matching gaps:

$$\frac{(A \rightarrow \alpha_1 \bullet \beta_1, i, j, \Gamma, \Upsilon_1 \oplus \langle C \rightarrow \alpha_1 \bullet D \beta_2 \rangle \oplus \Upsilon_2, \Psi_1 \oplus D \oplus \Psi_2)}{(A \rightarrow \alpha_1 \bullet \beta_1, i, j, \Gamma, \Upsilon_1 \oplus \langle C \rightarrow \alpha_1 D \bullet \beta_2 \rangle \oplus \Upsilon_2, \Psi_1 \oplus \Psi_2)}$$

Emptying Υ :

$$\frac{(A \rightarrow \alpha_1 \bullet, i, j, \Gamma, \Upsilon_1 \oplus \langle C \rightarrow \alpha_2 \bullet \rangle \oplus \Upsilon_2, \llbracket \rrbracket)}{(A \rightarrow \alpha_1 \bullet, i, j, \Gamma, \Upsilon_1 \oplus \Upsilon_2, \llbracket \rrbracket)} \text{root}(C)$$

Complete wrapping:

$$\frac{(A \rightarrow \alpha_1 \bullet, i, j, \Gamma, \langle C \rightarrow \alpha_2 \bullet \rangle, \llbracket \rrbracket)}{(C, i, j, \Gamma, \llbracket \rrbracket)} \neg \text{root}(C) \wedge \text{root}(A)$$

³We do not move any longer through productions with d-daughters in their rhs, these are processed as part of Υ in their target tree.

Appendix: Parsing with revised wrapping

Old rules that have to be kept:

$$\text{AX: } \frac{}{(N \rightarrow \bullet \alpha, i, i, [], [], [])} \quad i \in \{0, \dots, n-1\}$$

$N \rightarrow \alpha$ is a rule

$$\text{SC: } \frac{(N \rightarrow \alpha \bullet M \beta, i, j, \Gamma, \Upsilon, [])}{(N \rightarrow \alpha M \bullet \beta, i, j+1, \Gamma, \Upsilon, [])} \quad \ell(M) = s_{j+1}$$

$$\text{EM: } \frac{(N \rightarrow \alpha \bullet M \beta, i, j, \Gamma, \Upsilon, [])}{(N \rightarrow \alpha M \bullet \beta, i, j, \Gamma, \Upsilon, [])} \quad \ell(M) = \epsilon$$

$$\text{DE: } \frac{(N \rightarrow \alpha \bullet, i, j, \Gamma, [], [])}{(N, i, j, \Gamma, [])}$$

$$\text{SU: } \frac{(N \rightarrow \alpha \bullet M \beta, i, j, \Gamma_1, \Upsilon, []) \quad (R, j, k, \Gamma_2, [])}{(N \rightarrow \alpha M \bullet \beta, i, k, \Gamma_1 \oplus \Gamma_2, \Upsilon, [])} \quad \begin{array}{l} \text{leaf}(M) \\ \text{root}(R) \wedge \neg \text{sister}(R) \\ \ell(M) = \ell(R) \end{array}$$

$$\text{SA: } \frac{(N \rightarrow \alpha \bullet \beta, i, j, \Gamma_1, \Upsilon, []) \quad (M, j, k, \Gamma_2, [])}{(N \rightarrow \alpha \bullet \beta, i, k, \Gamma_1 \oplus \Gamma_2, \Upsilon, [])} \quad \ell(M) = \ell(N) \wedge \text{sister}(M)$$