

A workspace-based analysis of adjuncts

Daniel Milway
University of Toronto
dan.milway@gmail.com

Annual Meeting of the LSA
5 January 2020

I Overview

(P₁) THE PROBLEM

Adjuncts are best characterized as syntactically invisible/vacuous subexpressions.

(P₂) MY SOLUTION

This can be formalized if adjuncts are not attached to their “hosts”, but derived in parallel with those “hosts”.

(P₃) CORROBORATION

This formalization predicts a number of secondary properties of adjuncts.

I.1 Plan

1. Divide properties of adjuncts into primary and secondary.
2. Argue for (P₁) using the primary properties of adjuncts as evidence
3. Sketch out (P₂) in workspace-theoretic terms
4. Argue for (P₃).
5. Discuss some open questions that my proposal raises.

2 Primary/Secondary properties of Adjuncts

Adjuncts have a number of properties:

i. Freely ordered

- (1) a. Sadie sang the song with gusto after dinner.
b. Sadie sang the song after dinner with gusto.

ii. Optional

- (2) a. Sadie sang the song with gusto.
b. Sadie sang the song.

iii. Stackable

- (3) a. Sadie sang the song with gusto.
b. Sadie sang the song with gusto after dinner.

iv. Islands

- (4) *Who did Sadie invite Violet without meeting ____{wh}?

v. (but parasitic gaps)

- (5) Who did Sadie invite ____{wh} without meeting ____{PG}?

vi. Conjunctive interpretation

(2a) \approx Sadie sang the song **and** She did so with gusto.

vii. And more ...

- There’s an intuitive difference between the first three properties and the remainder.
 - The first three are somehow *essential* properties of adjuncts.
 - * These are the **primary properties**.
 - The remainder are things we discovered later.
 - * These are the **secondary properties**.
- This dichotomy suggests a method of theorizing:
 - Start by constructing a theory that *explicitly* captures the primary properties,
 - Then test that theory against the secondary properties.

3 Developing a theory of adjuncts

- Adjuncts are:
 1. Freely ordered
 2. Optional
 3. Stackable
- Can we get these down to a single property?

- (6) a. Sadie [sang the song]_α.
 b. Sadie [[sang the song]_α with gusto]_β.
 c. Sadie [[[sang the song]_α with gusto]_β after dinner]_γ.
 d. Sadie [[[[sang the song]_α after dinner]_δ with gusto]_ζ.
- These examples are equally grammatical.
 - Beyond that, the labeled expressions are *syntactically equivalent* to each other.
 - If $\alpha, \beta, \gamma, \delta,$ and ζ are all VPs, selected by, say, T, then any version of T that selects α will also select the other labeled VPs.
 - *cf* specifiers/complements
- (7) a. Joe hit the pillow.
 b. *Joe hit.
- (8) a. I believe Omar ate the pizza.
 b. *I believe ate the pizza.
- If a host-adjunct expression $H \hat{\ } A$ is grammatically equivalent to its host H, then the adjunct must be syntactically vacuous.
 - (P_I)
 - $H \hat{\ } A \equiv_{\sigma} H$

4 Formalizing (P_I) in derivational minimalism

- the core of derivational minimalism:
 - Phrases and sentences are derived by successive application of Merge.
- (9) Merge(X, Y) → {X, Y}
- The output of Merge is an expression distinct from its inputs.
 - {X, Y} ≠_σ X ≠_σ Y
 - Merge is not a good candidate for adjuncts, given (P_I)
 - Previous accounts introduce complications to the grammar to allow for adjuncts:
 - A new operation (Chomsky 2004)
 - An extra cycle of syntax (Stepanov 2001; Lebeaux 1991)
 - But adjunction is optional

Conjecture Any concept/thought expressed by a phrase/sentence with adjuncts can be expressed by a set of sentences/phrases without adjuncts.

- General considerations of theoretical parsimony militate against adding anything to our theory that we don't absolutely need.
- Instead, we'll make do with mechanisms we would need anyway:
 - Deletion of redundant structure
 - Workspaces

4.1 Deletion

- I will make the now-standard assumption that long-distance dependencies are created by merging single phrases in multiple positions.
 - Copy theory of movement
 - Only one “copy” of the phrase is pronounced.
 - The others must be deleted.
- (10) a. **Derived structure**
 {{The cake}{was eaten {the cake}}}
- b. **Delete copies**
 {{The cake} {was eaten {~~the cake~~}}}
- c. **Pronounce**
 The cake was eaten.
- Deletion seems to be governed by two factors:
 1. Identity
 - If two structures are identical, delete one.
 2. (Asymmetric) C-command Generally, delete the structurally lower phrase. (See Trinh (2009) for more details)

4.2 Workspaces

- Recent discussions of derivational minimalism have included the notion of *workspaces* (Collins and Stabler 2016; Chomsky 2019)
 - Generally, workspaces capture the intuition that arguments (usually NPs/DPs) are derived separately from the clausal spine.
- (11) Deriving *The 5 girls sang the anthem*
1. Derive *the anthem* in workspace 1
 2. Derive *the 5 girls* in workspace 2
 3. Derive the entire clause in workspace 3, which includes the result of 1 and 2
- Individual workspaces are encapsulated
 - The domain of Merge is the workspace
 - If a phrase is being derived in a workspace, all of its constituents must be included in that workspace.
- (12) WS₁: <the, girls>
 WS₂: <the, anthem>
- a. Merge(*the, girls*) → <{the, girls}>, <the, anthem>
 - b. Merge(*girls, anthem*) → undefined

4.3 The theory of adjuncts

4.3.1 First pass

- Adjuncts derived in separate workspaces which are never merged with their hosts.
 - unlike arguments which *are* merged with their predicates.

(13) Deriving *The 5 girls sang the anthem with gusto*

- Derive *the anthem* in WS₁
- Derive *the 5 girls* in WS₂
- Derive *with gusto* in WS₃
- Derive the entire clause in WS₄, which includes the result of 1 and 2

(14) $\langle\langle\{the, girls\}, \{pst, \{sing, \{the, anthem\}\}\}\rangle\rangle,$
 $\langle\{with, gusto\}\rangle$

- The result of this derivation is a pair of expressions, which we can linearize accordingly.
- Problem: Adjuncts seem to have scope
 - The visible visible stars* (Larson 1998)
 - She won't have danced on Sunday*
 - Cartography

(15) $\langle\{she \{not \{will, \{have \{dance\}\}\}\}\rangle\rangle,$
 $\langle\{on, Sunday\}\rangle$

- How can we differentiate the possible scopes of *on Sunday*?

4.3.2 Second Pass

- Consider how scope is treated in an X-bar theoretic phrase structure:
 - an adjunct's scope is its c-command domain
 - * A takes B in its scope if A C-commands B

- Now consider:

(16) $\langle\{she \{Neg \{T, \{Perf \{dance\}\}\}\}\rangle\rangle,$
 $\langle\{she \{Neg \{T \{Perf \{on, Sunday\}\}\}\}\rangle\rangle$

- Here, the PP doesn't "scope over" the verb,
- but now the PP and the verb scope under the same nodes
 - PP's c-commanders: $\{she, Neg, T, Perf\}$
 - dance*'s c-commanders: $\{she, Neg, T, Perf\}$
- How do we derive it?
 - The workspaces are independent up to a point ...
 - * VP for the host
 - * PP for the adjunct
 - After this point the two workspaces are derived in lockstep

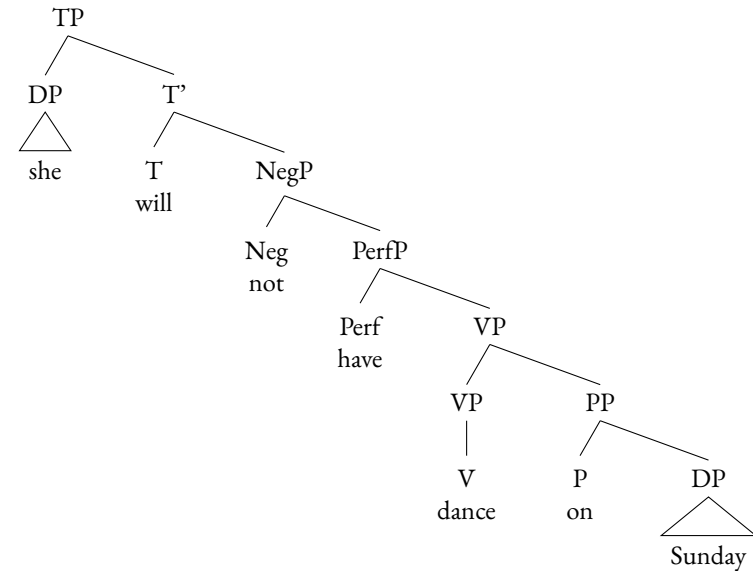


Figure 1: Low scope

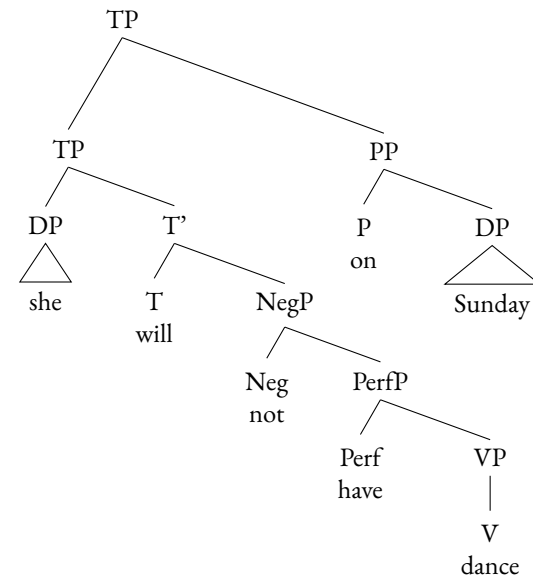


Figure 2: High scope

- * Every operation in one workspace, is mirrored in the other
- * When Merge(Perf, *dance*) occurs, so does Merge(Perf, *on Sunday*), and so on.
- Why don't we pronounce all the stuff above the PP?
 - It gets deleted
 - * It's identical to the stuff in the host
 - * Not c-commanded, but decidedly ordered
 - WS₁, WS₂ ≠ WS₂, WS₁

4.4 Interim Summary

- A Host-adjunct expression is underlyingly a pair of structures.
 - Each structure is derived in its own workspace.
 - These structures have identical “heads” and distinct “tails”.
- (17) <{*Sadie*, {T, {*sing*, {*the*, *song*}}}}>, <{*Sadie*, {T, {*with*, *gusto*}}}>
- The identical parts of the adjunct is deleted.
- (18) <{*Sadie*, {T, {*sing*, {*the*, *song*}}}}>, <{*Sadie*, {T, {*with*, *gusto*}}}>
- The workspaces are inherently ordered and this order is respected in pronunciation
 - Adjunction is syntactically vacuous because host-adjunct structures don't exist in the syntax.

5 Secondary Properties

5.1 Island-hood

- Under this theory, it follows directly from the fact that the domain of Merge is restricted to the workspace.
 - In (19) *who* cannot be merged with the host, because it is in a different workspace.
- (19) a. *Who did Sadie invite Violet without meeting ____{wh}?
- b. <{*Who* {C_Q, {... {*Sadie*, {*invite*, *Violet*}}}}}>, <{C, {... {*without* {*meeting*, *who*}}}}>
- If *who* moves within its workspace, we lose the identity portion of deletion for the adjunct.
- (20) <{C, {... {*Sadie*, {*invite*, *Violet*}}}}}>, <{*who* {C_Q, {... {*without* {*meeting*, *who*}}}}}>
- This might surface, but not as (19 a)

5.2 Parasitic Gaps

- Parasitic gaps occur when two parallel Wh-movement operations occur in separate workspaces

(21) a. Who did Sadie invite ____{wh} without meeting ____{wh}?

b. <{*Who* {C_Q, {... {*Sadie*, {*invite*, *who*}}}}}>, <{*who* {C_Q, {... {*without* {*meeting*, *who*}}}}}>

- Unlike (19), each *who* stays within its workspace
- Unlike (20), the higher *who* and C_Q in the adjunct can be deleted.

5.3 Interpreting host-adjunct structures

- Adjunction is (generally) characterized by a conjunctive interpretation.
 - Predicate Modification/Event Identification in standard formal semantics.
- In this theory host and adjunct are independent expressions.
- They compose like independent sentences:

(22) *The sky is blue.*, *The chair broke*
→ *the sky is blue* **and** *the chair broke*.

(23) <{*the*, *girls*}, {pst, {*sing*, {*the*, *anthem*}}}}>, <{*the*, *girls*}, {pst, {*with*, *gusto*}}}>
→ *the girls sang the anthem* **and** *the girls did so with gusto*.

- If the domain of Predicate Modification is coextensive with host-adjunct structures, then we can eliminate it from our repertoire of compositional operations.

6 Conclusions

6.1 The basic proposal

- Host-adjunct expressions are the result of two (or more) expressions being derived in parallel workspaces.
- No new mechanisms
 - Workspaces and deletion are needed anyway
- Existing mechanisms are not complicated
 - Merge is unchanged.
 - Delete is generalized
 - * Asymmetric c-command → any ordering.
- Naturally predicts adjunct islands, parasitic gaps, predicate modification

6.2 Possible extensions

- Coordination
 - Bošković (forthcoming) argues that the coordinated structure constraint can be unified with adjunct islands.
 - Chomsky (2019) analyzes both as results of pair-merge.
- Ellipsis
 - Adjunction:
 - * WS₁ and WS₂ are derived in parallel.
 - * Delete the head of WS₂
 - Ellipsis:
 - * WS₁ and WS₂ are derived in parallel.
 - * Delete the tail of WS₂?
- Head movement?
 - Also often taken to be pair-merge

6.3 Open Questions

- How is lockstep derivation ensured?
 - Generate and filter?
 - Some mechanism of controlling derivations?
- Non-adjunct “adjuncts”
 - Topicalized PPs, AdvPs, *etc*

References

Bošković, Željko. Forthcoming. “On Unifying the Coordinate Structure Constraint and the Adjunct Condition.” Edited by A. Bányai, T. Biberauer, J. Douglas, and S. Vikner. Language Science Press. https://boskovic.linguistics.uconn.edu/wp-content/uploads/sites/2801/2019/05/Boskovic_On-Unifying-the-Coordinate-Structure-Constraint-and-the-Adjunct-Condition.pdf.

Chomsky, Noam. 2004. “Beyond Explanatory Adequacy.” In *Structures and Beyond: The Cartography of Syntactic Structures*, 3:104–31. Oxford University Press.

———. 2019. *MIT Lecture on 12 April 2019*. https://www.dropbox.com/s/kihgbzpw2uzdo2/2976_Noam_Chomsky_Lecture_4-12.mp4.

Collins, Chris, and Edward Stabler. 2016. “A Formalization of Minimalist Syntax.” *Syntax* 19 (1): 43–78. <https://doi.org/10.1111/synt.12117>.

Larson, Richard K. 1998. “Events and Modification in Nominals.” In *Semantics and Linguistic Theory*, 8:145–68.

Lebeaux, David. 1991. “Relative Clauses, Licensing, and the Nature of the Derivation.” *Syntax and Semantics* 25: 209–39.

Stepanov, Arthur. 2001. “Late Adjunction and Minimalist Phrase Structure.” *Syntax* 4 (2). Wiley Online Library: 94–125.

Trinh, Tue. 2009. “A Constraint on Copy Deletion.” *Theoretical Linguistics* 35 (2-3). Walter de Gruyter GmbH & Co. KG: 183–227.