

# Subjects of Adjuncts and Labeling

Dan Milway

Dog Days Workshop  
17 August 2016

## 1 Introduction

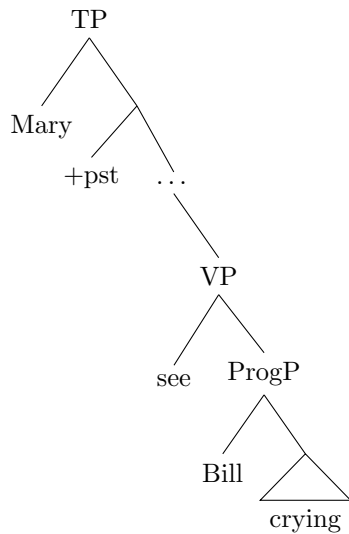
- ACC-ing clauses (ACs) are non-finite clauses with accusative subjects and progressive main verbs.
  - Often embedded in direct perception reports.
- (1)    a.    They saw her proving the theorem.  
       b.    She was seen proving the theorem.
- Subjects of ACs have a strange movement pattern.
    - Subjects of pseudo-relatives (PRs) seem to show the same pattern. (Cinque 1996)
  - The pattern can be partially accounted for by assuming two label-based theories:
    - Labels are assigned algorithmically as a requirement of the Conceptual-Intentional (CI) interface. (Chomsky 2013, 2015)
    - Adjunction structures do not receive labels (Chametzky 1996; Hornstein 2009)
  - These assumptions require a rethinking of the CI interface which can allow for a full account of the pattern

## 2 The phenomenon: ACC-ing (and Pseudo-relative?) subjects

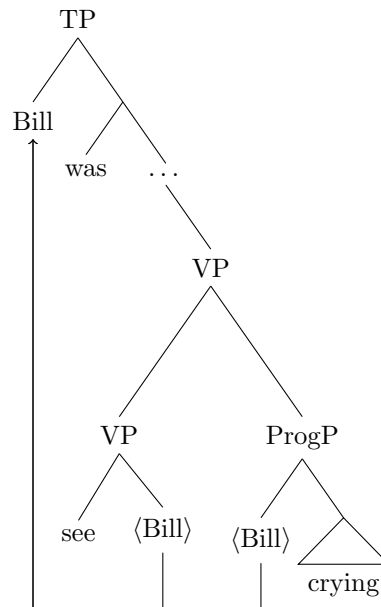
- In active sentences, the perception verb doesn't  $\theta$ -mark the ACC-ing subject.
  - In passives, the perception verb does  $\theta$ -mark the ACC-ing subject
- (2)    a.    We heard it raining last night.  
       b.    We saw all hell breaking loose.  
       c.    We heard Jamie being slandered.
- (3)    a.    \*It was heard raining last night.  
       b.    \*All hell was seen breaking loose. (\*idiomatic)  
       c.    \*Jamie was heard being slandered.

- THEME-marking occurs in Comp V.
- In actives, the AC occupies Comp V.
- In passives, the AC subject occupies Comp V.
  - The AC is adjoined to VP.

(4) **Active/Argument ACC-ing**

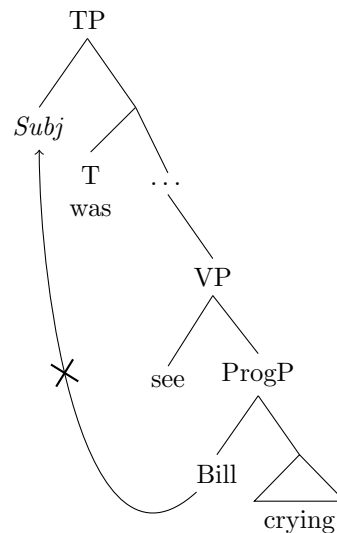
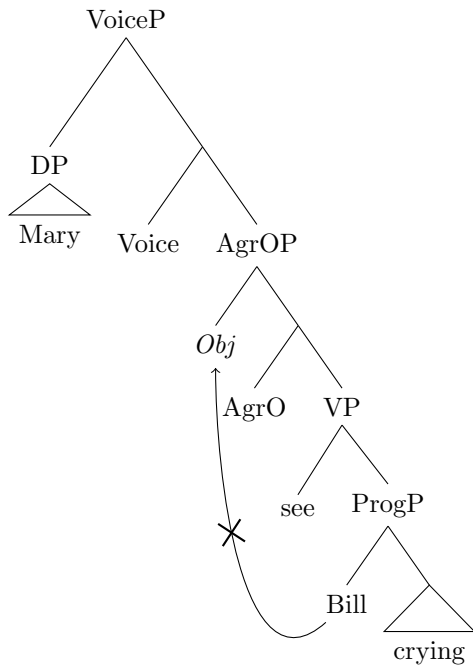


(5) **Passive/Adjunct ACC-ing**

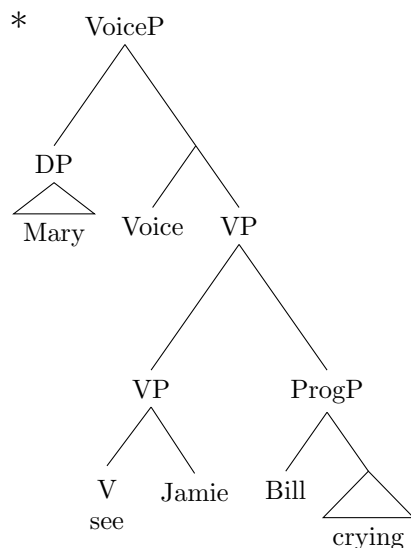


- Strange result: **The ACC-ing subject is frozen in the Argument ACC-ing, but must move from the Adjunct ACC-ing.**
- cp. Raising-to-Object and Adjunct Islands

(6) **Argument AC subjects do not move**



(7) **Adjunct AC subjects must move**



### 3 Label theory (Chomsky 2013, 2015)

#### 3.1 The Theory

- The narrow syntax is (simplest) Merge
  - $\text{Merge}(\alpha, \beta) = \{\alpha, \beta\}$
- Since Merge doesn't specify the label of its output, and the narrow syntax is only Merge, labels must be determined at one of the interfaces.
- Specifically: The CI interface.
  - Chomsky's CI primacy conclusion
- Labels are assigned by a special instance of Minimal Search, the Labelling Algorithm (LA), upon Transfer at the phase level.
- Unlabellable objects cause a crash.

But:

- No current theory of semantics has any need for labels.
  - For type-driven interpretation, only the content of syntactic objects is required
  - For a neo-Davidsonian theory, only the merge order of arguments is required
- If our current understanding of the syntax-semantics interface is correct, the proposal above must be wrong.
- **$\therefore$  If the proposal above is correct, our current understanding of the syntax semantics interface is wrong.**

### 3.2 Labelling algorithm

- LA, when applied to a syntactic object SO, searches SO for its most prominent sub-object and assigns that as SO's label.
- Chomsky (2013) discusses the three logical possibilities:

- (8)
- $LA(\{X, YP\}) = X$
  - $LA(\{X, Y\}) = \begin{cases} X & \text{if } Y \text{ is a root, and } X \text{ is not a root} \\ \text{Undefined} & \text{otherwise} \end{cases}$
  - $LA(\{XP, YP\}) = \begin{cases} \langle F, F \rangle & \text{if } XP \text{ and } YP \text{ agree for some feature } F \\ LA(YP) & \text{if } XP \text{ is a lower copy in a chain} \\ \text{Undefined} & \text{otherwise} \end{cases}$

## 4 Labelling explanation

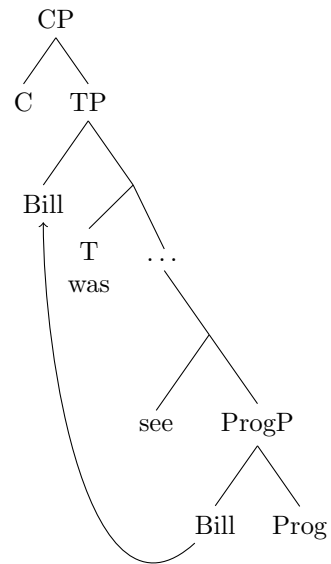
- |                         |                         |
|-------------------------|-------------------------|
| (9) <b>Argument ACs</b> | (10) <b>Adjunct ACs</b> |
| a. $\{DP, ProgP\}$      | a. $*\{DP, ProgP\}$     |
| b. $*\{t, ProgP\}$      | b. $\{t, ProgP\}$       |

### 4.1 Argument ACC-ing subjects cannot move

- Subject of argument ACs show **Criterial freezing** (in Rizzi's terms).
- Chomsky (2015) proposes a labelling account for this type of freezing.
  - $\{XP_F, \{Y_F, ZP\}\}$  is labelled  $\langle F, F \rangle$ .
  - Y is “too weak” to label on its own.
  - $\{t, \{Y_F, ZP\}\}$  is unlabellable so it yields a crash.
- Replace XP with the AC subject and Y with  $Prog^0$  and we have our account.

- (11)
- $*Bill_i$  was see-en [  $t_{see}$  [  $t_i$  throwing the ball]]
  - Derivation**

- 1.Merge(Bill,  $\{Prog, YP\}$ )
- 2.Transfer+Label( $YP$ )<sup>1</sup>
- 3.Merge(see, ProgP)
- ...
- (Derive the finite clause)
- ...
- 4.(Internal-)Merge(Bill, T')
- 5.Merge(C, TP)
- 6.Transfer+Label(TP)
- \*CRASH\*
- ( $\{t, Prog\}$  is unlabellable.)



<sup>1</sup>Assuming  $Prog^0$  is a phase head, following Harwood (2015)

## 4.2 Adjunct ACC-ing subjects can move

- **Assumption:** Adjunction structures do not receive a label. (Chametzky 1996; Hornstein 2009)
  - If  $\{XP, YP\}$  is an adjunction structure, LA skips it, and moves on to the adjunction host.
- The AC is an adjunct and thus, invisible the labelling algorithm.
- It follows that the internal structure of the AC is also invisible to LA.
- $\{t, \text{Prog}\}$  is still unlabellable, but doesn't lead to a crash.
  - Crashes occur when LA fails

## 4.3 Adjunct ACC-ing subjects *must* move.

- If labelling is required at the CI interface, it must have some semantic potency.
- So  $\{XP, YP\}$  will be interpreted differently depending on its label
  - Criterial:  $\text{Label}(\{XP, YP\}) = \langle F, F \rangle \rightarrow \text{Abstraction}$ 
    - \* Including, but not limited to, lambda abstraction (cp Lohndal and Pietroski 2011)
  - Adjunct:  $\text{Label}(\{XP, YP\}) = \emptyset \rightarrow \text{Conjunction}$
- An unlabelled  $\{DP, \{\text{Prog}, YP\}\}$  is interpreted as the conjunction of a ProgP predicate and its subject.
  - This is (likely) a deviant interpretation
- An unlabelled  $\{t, \{\text{Prog}, YP\}\}$ , however, does not yield a deviant interpretation.
  - This is stipulated for now.

## 5 “Conclusion”

- Chomsky proposes that for a derivation to converge at CI it must produce a labellable syntactic object.
- If Chomsky is right, we need to rethink our conception of the syntax-semantics interface.  
**My Proposal:** The label of a syntactic object has consequences for that object's interpretation.
- I have shown how a puzzling fact about ACC-ing subjects can be straightforwardly explained, given Chomsky's proposal (and my extension).
- Several aspects require more explanation/work:
  - How does the LA “know” to skip adjunction structures?
  - How are  $\{t, XP\}$  structures interpreted?
  - I predict that no adjunct phrases should have criterial specifiers.

## References

- Chametzky, Robert (1996). *A theory of phrase markers and the extended base*. Albany: SUNY Press.
- Chomsky, Noam (2013). “Problems of projection”. In: *Lingua* 130, pp. 33–49.
- (2015). “Problems of projection: Extensions”. In: *Structures, Strategies and Beyond: Studies in honour of Adriana Belletti*. Ed. by Elisa Di Domenico and Simona Matteini Cornelia Hamann. Philadelphia: John Benjamins, pp. 3–16.
- Cinque, Guglielmo (1996). “The pseudo-relative and ACC-ing constructions after verbs of perception”. In: *Italian syntax and Universal Grammar*. Cambridge: Cambridge University Press, pp. 244–275.

- Harwood, William (2015). “Being progressive is just a phase: celebrating the uniqueness of progressive aspect under a phase-based analysis”. In: *Natural Language & Linguistic Theory* 33.2, pp. 523–573.
- Hornstein, Norbert (2009). *A theory of syntax: minimal operations and universal grammar*. Cambridge: Cambridge University Press.
- Lohndal, Terje and Paul Pietroski (2011). “Interrogatives, instructions, and I-languages: An I-semantics for questions”. In: *Linguistic Analysis* 37.3-4, pp. 458–515.