Raising in LCG

Carl Pollard

December 4, 2014

Starting to Get Real: English 'NPs'

- To get started, we assumed tectos NP (for names) and It (for dummy *it*), but this is too simple.

- Even if we consider only third person singular noun phrases, we still must account for these facts:

  - Names and NPs formed by combining a determiner with a common noun occur both as subject and as object of verb or preposition.
  - The same is true of the dummy pronoun.
  - But, except for nonhuman *it*, definite pronouns have different forms, of which some (*he, she*) can't be objects and others (*her, him*) can't be subjects.
  - Only a few verbs, e.g. *be*, *seem*, and weather verbs, allow dummy subjects.
  - And only a few verbs, e.g. *believe* (as in *believe it to be raining* allow dummy objects.

Are Features Necessary?

- In most syntactic frameworks (CCG, HPSG, LFG, MP) problems of this kind are addressed through the use of **features**.

- For example, in HPSG, NPs specify values for the features CASE and NFORM.

- But in a framework based on proof theory, it's unclear what 'features' would be: formulas aren't usually thought of as having 'features'.

- We'll use a different approach due to Lambek (1999) in the context of his framework called **pregroup grammar**.

- Pregroup grammar is based on *bilinear* logic, but this idea works just as well with linear logic.

Ordering the Basic Tectos (1/2)

- Lambek proposed *ordering* the basic syntactic types.

- The basic intuition is that if $A \leq B$, then any sign with tecto A can also be considered as a sign with tecto B.

- In this case we say A is a **(tecto) subtype** of B.

- For example: we would like to say that the tecto of 'NPs' which can serve as both subjects and objects (which we will call Neu, for 'neutral') is a subtype of the tecto of 'NPs' that can serve as subjects (which we will call Nom, for 'nominative'):

$$\text{Neu} \leq \text{Nom}$$

Ordering the Basic Tectos (2/2)

- In the grammar, we directly assert certain inequalities, such as Neu $\leq$ Nom, and then define $\leq$ to be the smallest order on basic tectos that includes all the asserted inequalities.

- Then we revise the Trace axiom schema to the following more general form (the original schema corresponds to the case $B = B'$):

  Trace Axiom Schema (Generalized):

$$x; B; z \vdash x; B'; z \text{ (for } B \leq B')$$

Three Derived Rule Schemas

These schemas (schematized over $B \leq B'$) are useful for shortening LCG proofs. (Their derivations are left as exercises.)

- Derived Rule Schema 1

$$\frac{\Gamma \vdash a; B; c}{\Gamma \vdash a; B'; c}\,\text{D1}$$

- Derived Rule Schema 2

$$\frac{\Gamma \vdash f; B' \multimap A; g}{\Gamma \vdash f; B \multimap A; g}\,\text{D2}$$

- Derived Rule Schema 3

$$\frac{\Gamma \vdash f; A \multimap B; g}{\Gamma \vdash f; A \multimap B'; g}\,\text{D3}$$

Ordering Basic Tectos to Analyze English Case

- For now we only consider sentences with finite verbs.
- Later we'll elaborate our approach to handle issues about 'unrealized' subjects of nonfinite verb forms (base forms, infinitives, and participles) and of nonverbal predicative expressions.
- First we discard the tecto NP and replace it with:
  - Nom ('NPs' that can be subjects of finite verbs)
  - Acc ('NPs' that can be objects of verbs or prepositions)
  - Neu ('NPs' that can be either)
- Next, we assert the inequalities

$$\text{Neu} \leq \text{Nom}, \text{Neu} \leq \text{Acc}$$

Lexicon Revised and Expanded to Analyze Case

(Semantics omitted from now on, until we need it)

$\vdash \mathrm{pedro}; \mathrm{Neu}$

$\vdash \mathrm{chiqita}; \mathrm{Neu}$

$\vdash \mathrm{maria}; \mathrm{Neu}$

$\vdash \mathrm{she}; \mathrm{Nom}$

$\vdash \mathrm{he}; \mathrm{Nom}$

$\vdash \mathrm{him}; \mathrm{Acc}$

$\vdash \mathrm{her}; \mathrm{Acc}$

$\vdash \lambda_s.s \cdot \mathrm{brayed}; \mathrm{Nom} \multimap \mathrm{S}$

$\vdash \lambda_{st}.s \cdot \mathrm{believed} \cdot t; \mathrm{Nom} \multimap \bar{\mathrm{S}} \multimap \mathrm{S}$

$\vdash \lambda_{st}.s \cdot \mathrm{beat} \cdot t; \mathrm{Nom} \multimap \mathrm{Acc} \multimap \mathrm{S}$

$\vdash \lambda_{stu}.s \cdot \mathrm{gave} \cdot t \cdot u; \mathrm{Nom} \multimap \mathrm{Acc} \multimap \mathrm{Acc} \multimap \mathrm{S}$

How Neutral ‘NPs’ Get Case

This derivation uses Derived Rule Schema 1 twice:

$$\dfrac{\dfrac{\vdash \lambda_{st}.s \cdot \mathrm{beat} \cdot t; \mathrm{Nom} \multimap \mathrm{Acc} \multimap \mathrm{S} \qquad \dfrac{\vdash \mathrm{pedro}; \mathrm{Neu}}{\vdash \mathrm{pedro}; \mathrm{Nom}}}{\vdash \lambda t.\mathrm{pedro} \cdot \mathrm{beat} \cdot t; \mathrm{Acc} \multimap \mathrm{S}} \qquad \dfrac{\vdash \mathrm{chiquita}; \mathrm{Neu}}{\vdash \mathrm{chiquita}; \mathrm{Acc}}}{\vdash \mathrm{pedro} \cdot \mathrm{beat} \cdot \mathrm{chiquita}; \mathrm{S}}$$

Predicative Adjectives

- As a first approximation, we analyze predicative adjectives with a new basic tectotype PrdA:

  $\vdash \mathrm{lazy}; \mathrm{PrdA}$

  $\vdash \mathrm{asleep}; \mathrm{PrdA}$

- We can’t do anything with these yet, but we are about to fix that.

Introducing the Copula *Be*

- As a first approximation, *be* takes a noun phrase subject, which for finite forms of *be* must be nominative, and a predicative adjective complement (actually, there other kinds of predicatives besides adjectives are possible, which we ignore for now):

  $\vdash \lambda_{st}.s \cdot \text{is} \cdot t; \text{Nom} \multimap \text{PrdA} \multimap \text{S}$

- Problem: Some PrdAs demand a dummy *it* subject, while most require a 'normal', nondummy, subject:

  1. Chiquita/He/She is lazy/asleep.
  2. $*$ Chiquita/He/She is rainy.
  3. It is rainy.
  4. $*$ It is lazy/asleep. (where *it* is not referential)

  How does the copula know what kind of subject its complement expects?

Predicative Adjectives 'Care' about their Subjects

- Although a predicative adjective cannot directly take a subject, if a copula takes it as a complement, it 'tells' the copula what kind of subject to take.

- We analyze this by treating predicative adjectives **tectogrammatically** (and semantically) as functors, but phenogrammatically as just strings:

  $\vdash \text{rainy}; \text{It} \multimap \text{PrdA}$

  $\vdash \text{obvious} : \bar{\text{S}} \multimap \text{PrdA}$

  $\vdash \text{lazy} : \text{Nom} \multimap \text{PrdA}$

  The 'Nom' in the last entry is not quite right, but it will take some development to see why.

- We will analyze nonfinite verb phrases (infinitivals, base-form verb phrases, and participial phrases) the same way, but with PrdA replaced by other basic tectos (Inf, Bse, Prp, Psp, and Pas).

Be, Take Two

- Now, we replace our old lexical entry for *is*:

  $\vdash \lambda_{st}.s \cdot \text{is} \cdot t; \text{Nom} \multimap \text{PrdA} \multimap \text{S}$

  with the following schema:

  $\vdash \lambda_{st}.s \cdot \text{is} \cdot t; A \multimap (A \multimap \text{PrdA}) \multimap \text{S}$

  where A is a metavariable ranging over tectos.

- This analysis corresponds to what is called **raising to subject (RTS)** in other frameworks.

- In essence, *is* says: 'I don't care what my subject is, as long as my complement is happy with it'.

- We use the same trick to analyze other verbs (and nonverbal predicatives) traditionally analyzed in terms of RTS (e.g. modals and other auxiliaries, *seem*, *tend*).

Problems with Raising (1/2)

- Another problem: some verbs, traditionally called **raising to object (RTO)** verbs, feel the same way about their object as RTS verbs feel about their subject, for example *considers*:

  1. Pedro considers it rainy.
  2. Pedro considers that Chiquita brays obvious.
  3. Pedro considers Chiquita/her/∗she lazy.

- For such verbs, if the object is a pronoun, it has to be *accusative*.

Problems with Raising (2/2)

So if we try to analyze RTO on a par with RTS, with a lexical entry like:

$\vdash \lambda_{stu}.s \cdot \text{considers} \cdot t \cdot u; \text{Nom} \multimap A \multimap (A \multimap \text{PrdA}) \multimap \text{S}$

it will interact badly with the lexical entry

$\vdash \text{lazy} : \text{Nom} \multimap \text{PrdA}$

to overgenerate things like

$*$ Pedro considers she lazy.

while failing to generate the correct

Pedro considers her lazy.

Fixing the Undergeneration Problem with Raising (1/2)

- The undergeneration problem arises with RTO because the lexical entries for predicative adjectives like *lazy* demand *nominative* subjects.

- This works when the 'unrealized' subject is 'raised' to the subject of a finite verb (such as *is*), but not when it is 'raised' to object, where an *accusative* is needed.

- We could add a second entry with tecto Acc $\multimap$ PrdA.

- But we can avoid doubling up all these lexical entries if instead we replace all the Nom $\multimap$ PrdA entries with entries with tecto PRO $\multimap$ PrdA, where PRO is a new basic tecto ordered as follows:

$$\text{Nom} \leq \text{PRO}, \text{Acc} \leq \text{PRO}$$

Fixing the Undergeneration Problem with Raising (2/2)

- Then in the lexicon we need only list

  $\vdash \text{lazy}; \text{PRO} \multimap \text{PrdA}$

- From this we can derive the signs needed as complements to *is* and *considers*, respectively, by Derived Rule Schema 2:

  $\vdash \text{lazy}; \text{Nom} \multimap \text{PrdA}$

  $\vdash \text{lazy}; \text{Acc} \multimap \text{PrdA}$

- While Neu is *overspecified* between Nom and Acc, PRO is *underspecified* between Nom and Acc.

- Cf. GB theory's PRO, which is supposed to occur in non-case-assigned positions such as subject of infinitive.

- But unlike GB, our predicatives (and nonfinite VPs) don't actually *take* subjects, because phenogrammatically they are not functions.

Fixing the Overgeneration Problem with Raising (1/2)

- As it stands, our analysis still *over*generates:

  1. $*$ Pedro considers she lazy.
  2. $*$ Her is lazy.

  because the *A*s in the lexical schemas for *is* and *considers* can be instantiated (inter alia) as Nom or Acc.

- Our *is* doesn't care what its subject is as long as its complement likes it, and our *considers* doesn't care what its object is as long as its complement likes it.

- But *is should* insist that if its subject is a (nondummy) NP, then it must be nominative.

- And *considers should* insist that if its object is a (nondummy) NP, then it must be accusative.

Fixing the Overgeneration Problem with Raising (2/2)

- We solve these problems by limiting the possible instantiations of the type variable A in the lexical entries, in different ways.
- We add two new basic tectotypes NOM and ACC.
- NOMs are things that can be subjects of finite RTS verbs.
- ACCs are things that can be objects of RTO verbs.
- Next we add more tecto inequalities:

$$\mathrm{Nom} \leq \mathrm{NOM},\ \mathrm{It} \leq \mathrm{NOM},\ \mathrm{Acc} \leq \mathrm{ACC},\ \mathrm{It} \leq \mathrm{ACC}$$

- And finally, we revise the lexical schemas for *is* and *considers* as follows:

$\vdash \lambda_{st}.s \cdot \mathrm{is} \cdot t; A \multimap (A \multimap \mathrm{PrdA}) \multimap \mathrm{S}\ (A \leq \mathrm{NOM})$

$\vdash \lambda_{stu}.s \cdot \mathrm{considers} \cdot t \cdot u; \mathrm{Nom} \multimap A \multimap (A \multimap \mathrm{PrdA}) \multimap \mathrm{S}\ (A \leq \mathrm{ACC})$

Subjects of Nonfinite Verbs (1/3)

- As we've seen, the tecto requirement for subjects of predicatives and nonfinite verbs whose finite counterpart would require a Nom is PRO.
- And the tecto requirement for subjects of finite RTS verbs is NOM.
- But what is the type requirement for the subject of a *non*finite RTS verb, such as *be* or *to*? It is less constrained than objects of RTO verbs or subjects of finite RTS verbs, because *no* case requirement is imposed on it.
- We handle this by positing a new tecto, called NP (because it plays a role analogous to that of NP-trace in GB theory), of which NOM, PRO, and ACC are subtypes:

$$\mathrm{NOM} \leq \mathrm{NP},\ \mathrm{PRO} \leq \mathrm{NP},\ \mathrm{ACC} \leq \mathrm{NP}$$

Subjects of Nonfinite Verbs (2/3)

- Finally, we write lexical entries schematized over values of A which are subtypes of NP:

  $\vdash \lambda_s.\text{be} \cdot s; (A \multimap \text{PrdA}) \multimap A \multimap \text{Bse}\ (A \leq \text{NP})$

  $\vdash \lambda_s.\text{to} \cdot s; (A \multimap \text{Bse}) \multimap A \multimap \text{Inf}\ (A \leq \text{NP})$

- In the preceding lexical entries, the tectos are written with the complements as the **intial** arguments and the subject (which cannot be taken directly as an argument) **last**.
- This same practice is followed for all nonfinite verbs (and complement-taking nonverbal predicatives). Compare:

  $\vdash \lambda_{st}.s \cdot \text{beats} \cdot t; \text{Nom} \multimap \text{Acc} \multimap \text{S}$

  $\vdash \lambda_s.\text{beat} \cdot s; \text{Acc} \multimap \text{PRO} \multimap \text{Bse}$

Subjects of Nonfinite Verbs (3/3)

- Although verbs (other than *to*) don't have infinitive forms, roughly that effect results from syntactic combination:

  $$\dfrac{\dfrac{\lambda_s.\text{to} \cdot s; (A \multimap \text{Bse}) \multimap A \multimap \text{Inf}}{\lambda_s.\text{to} \cdot s; (\text{PRO} \multimap \text{Bse}) \multimap \text{PRO} \multimap \text{Inf}} \qquad \vdash \text{bray}; \text{PRO} \multimap \text{Bse}}{\vdash \text{to} \cdot \text{bray}; \text{PRO} \multimap \text{Inf}}$$

  Here for expository purposes we pretend that instantiation of a schema is a unary rule (of course it isn't really.)