Control and *Tough*–‘Movement’

Carl Pollard

Department of Linguistics
Ohio State University

February 2, 2012
We saw that PRO is used for the unrealized subject of nonfinite verbals and predicatives where the subject ‘plays a semantic role’ (and so dummy subjects are disallowed).

If such an expression is the complement of

- a RTO verb, then the PRO is ‘identified with’ the upstairs Acc object (here indicated informally by subscripts)
  [consider her₁ [PRO₁ conservative]]
- a finite RTS verb, then the PRO is ‘identified with’ the upstairs Nom subject
  [he₁ seems [PRO₁ conservative]]
- a nonfinite RTS verb or predicative, then the PRO is ‘identified with’ the upstairs PRO subject.
  [PRO₁ be [PRO₁ conservative]]

In all these cases, the upstairs object or subject identified with the PRO complement subject plays no semantic role with respect to the upstairs verbal/predicative.
But expressions with a PRO subject requirement are not always complements of raising verbs. For example, they can themselves be subjects, as in *to err is human*. Here the property of being human is being predicated of another property, the property of erring.

Such expressions can also be complements of a verb (or predicative), which (in a sense to be made precise) ‘identifies’ the unrealized downstairs subject *semantically* with one of its own arguments (either the subject or the object) which *does* play a semantic role upstairs.
Examples:

1. Chiquita tried to sing.
2. Pedro persuaded Chiquita to sing.

Verbs like these are often analyzed as describing a relation between one or two entities and a proposition about one of those entities (in the examples above, the proposition about Chiquita that she sings).

That entity (here, Chiquita), or the corresponding upstairs argument position (subject of *tried* or object of *persuaded*), is said to *control* the PRO subject of the complement.

In such cases the higher verb is called a *control* verb (and likewise for predicatives).
Control verbs are also called equi verbs because in early TG they were analyzed by a transformation (‘equi-NP deletion’) that deleted the complement subject (which was assumed to be identical with the controller).

By comparison, raising verbs in TG were analyzed by a different transformation (‘raising’) that moved the complement subject to a higher position in the tree.
In LG, the analysis of control makes no tectogrammatical connection between the complement subject and the controller, instead handling the connection semantically:

\[ \lambda_{st} \cdot s \cdot \text{tries} \cdot t; \text{Nom} \rightarrow (\text{PRO} \rightarrow \text{Inf}) \rightarrow S; \lambda_{xP} \cdot \text{try} \; x \; (P \; x) \]

\[ \lambda_{stu} \cdot s \cdot \text{persuaded} \cdot t \cdot u; \text{Nom} \rightarrow \text{Acc} \rightarrow (\text{PRO} \rightarrow \text{Inf}) \rightarrow S; \lambda_{xyP} \cdot \text{persuade} \; x \; y \; (P \; y) \]

Alternatively, control verb meanings can be treated as relations between one or two entities and a property:

\[ \lambda_{st} \cdot s \cdot \text{tries} \cdot t; \text{Nom} \rightarrow (\text{PRO} \rightarrow \text{Inf}) \rightarrow S; \lambda_{xP} \cdot \text{try} \; x \; P \]

\[ \lambda_{stu} \cdot s \cdot \text{persuaded} \cdot t \cdot u; \text{Nom} \rightarrow \text{Acc} \rightarrow (\text{PRO} \rightarrow \text{Inf}) \rightarrow S; \lambda_{xyP} \cdot \text{persuade} \; x \; y \; P \]

with the relationship between the controller the property captured via nonlogical axioms (‘meaning postulates’) of the semantic theory.
Paradigms like the following have troubled generative grammarians since the mid 1960s:

a. It is easy (for Mary) to please John.

b. John\textsubscript{i} is easy (for Mary) to please t\textsubscript{i}.

- The two sentences mean the same thing: that pleasing John is something that one (or Mary) has an easy time doing.
- It’s the (b) version that has been troublesome, because the object of the infinitive, indicated by t, seems to have moved to the subject position of the finite sentence.
- But the syntactic relationship, indicated by coindexation, between the object “trace” and the subject doesn’t fall straightforwardly under recognized rule types in the mainstream generative grammar tradition.
As expressed by Hicks (2009), citing Holmberg (2000): ‘Within previous principles-and-parameters models, TCs [tough constructions] have remained “unexplained and in principle unexplainable” because of incompatibility with constraints on θ-role assignment, locality, and Case.’

Hicks, building on a notion of “smuggling” introduced by Collins (2005), proposes a phase-based minimalist analysis in terms of “A-moving a constituent out of a ‘complex’ null operator that has already undergone Ā-movement.”
GB (early 1980’s) posited four kinds of EC’s

- (little) pro, essentially inaudible definite pronouns, not relevant for the present discussion
- trace (aka ‘syntactic variable’)
- NP-trace
- (big) PRO

These last three figured, respectively, in the analysis of:

- wh-movement, later subsumed under $\bar{A}$-movement (wh-questions, relative clauses, topicalization, clefts, pseudoclefts, etc.)
- NP-movement, later called A-movement (passive, raising)
- control
The theoretical assumptions about how these three kinds of empty elements worked never seemed to add up to a consistent story about TCs.

In LG we have counterparts of all three.

In due course we’ll see how LG fares in accounting for TCs.

First a glance at how the GB EC’s were supposed to work.
Wh-Movement/¯A-Movement

Something moves, possibly long-distance, from a Case-assigned, θ-role-assigned A-position to an ¯A position:

1. Who_o[t_i came]?
2. Who_o did [Mary see t_i]?
3. Who_o did [Mary say [John saw t_i]]? (long-distance)
4. * Who_o[t_i rained]? (launch site is non-θ)
5. * Who_o did [John try [t_i to come]]? (launch site is non-Case)
6. * Mary told John_o [she liked t_i]. (landing site is an A-position)

A = argument (subject or object)
¯A = nonargument
[...] = sentence boundary
Something moves from a non-Case, A-position to a superjacent, non-θ, A-position:

1. John$_i$ seems [n$_i$ to be happy].
2. It$_i$ seems [n$_i$ to be raining].
3. * John$_i$ seems [n$_i$ is happy]. (launch site is Case-assigned)
4. * John$_i$ seems [Mary believes [n$_i$ to be happy]]. (landing site is not superjacent)
5. * It$_i$ tries [n$_i$ to be raining]. (landing site is θ-assigned)
6. * Who$_i$ does [John seem [n$_i$ to be happy]]? (landing site is an Ā-position)
An EC in a $\theta$-assigned non-Case position is anaphoric to something in a superjacent $A$-position:

1. Mary$_i$ tries [PRO$_i$ to be happy].
2. * Mary$_i$/it$_i$ tries [PRO$_i$ to rain]. (EC is in a non-$\theta$ position.)
3. * John tries [Mary to like PRO$_i$]. (EC is in a Case position)
4. * Mary$_i$ tries [John believes [PRO$_i$ to be happy]]. (landing site is not superjacent)
5. * Who$_i$ did [John try [PRO$_i$ to be happy]]? (landing site is an $\bar{A}$-position)
What’s Tough about *Tough*-‘Movement’

- Like \(\bar{A}\)-movement, the launch site is a \(\theta\)-assigned Case position, and it can be long-distance:
  a. John\(_i\) is easy for Mary [to please \(t_i\)].
  b. John\(_i\) is easy for Mary [to get other people [to distrust \(t_i\)]].

- Like A-movement, the landing site is a non-\(\theta\) A-position.

- Like Control, the ‘antecedent’ of the EC must be ‘referential’, i.e. it can’t be a dummy or an idiom chunk:
  a. John is easy to believe to be incompetent.
  b. * It is easy to believe to be raining.
  c. * There is easy to believe to be a largest prime number.
  d. * The shit is easy to believe to have hit the fan. (no idiomatic interpretation)
Nom (nominative, e.g. *he*, *she*)
Acc (accusative, e.g. *him*, *her*)
For (nonpredicative *for*-phrase, e.g. *for Mary*)
It (‘dummy pronoun’ *it*)
S (finite clause)
Inf (infinitive clause)
Bse (base clause)
Prd (predicative clause)
PrdA (adjectival predicative clause)
Basic Tectos Involved in Analysis of TCs (2/2)

- Neu (case-neutral, e.g. John, Mary)

- PRO (LG counterpart of GB’s PRO)
  Used for subject of nonfinite verbs and predicatives that assign a semantic role to the subject, e.g. nonfinite please

- NP (LG counterpart of GB’s NP-trace)
  Used for subject of nonfinite verbs and predicatives that don’t assign a semantic role to the subject, e.g. nonfinite seem, infinitive to

- NOM (generalized nominatives)
  Used for subject of finite verbs that don’t assign a semantic role to the subject, e.g. seems, is

- ACC (generalized accusatives)
  Used for objects of verbs that don’t assign a semantic role to the object, e.g. infinite-complement-believe
Review of Basic Tecto Ordering

Neu < Nom
Neu < Acc
Nom < PRO
Acc < PRO
Nom < NOM
Acc < ACC
It < NOM
It < ACC
NOM < NP
ACC < NP
PRO < NP
PrdA < Prd
Some Nonlogical Constants for Lexical Semantics

⊢ j : e (John)
⊢ m : e (Mary)
⊢ rain : p
⊢ please : p₂
(The first argument is pleasing and the second argument experiences the pleasure.)
⊢ easy : e → p₁ → p
(The first argument is the one who has an easy time of it, and the second argument is the ‘piece of cake’.}

Carl Pollard  Control and Tough-‘Movement’
Lexical Entries

⊢ it; It; * (dummy pronoun it)
⊢ john; Neu; j
⊢ mary; Neu; m
⊢ \lambda_{st}.s \cdot pleases \cdot t; Nom \rightarrow Acc \rightarrow S; please
⊢ \lambda_{t}.please \cdot t; Acc \rightarrow PRO \rightarrow Bse; \lambda_{yx}.please x y
⊢ \lambda_{t}.to \cdot t; (A \rightarrow Bse) \rightarrow A \rightarrow Inf; \lambda_{P}.P (A \leq NP, P : B \rightarrow p)
⊢ \lambda_{st}.s \cdot is \cdot t; A \rightarrow (A \rightarrow Prd) \rightarrow S; \lambda_{xP}.P x (A \leq NOM, x : B, P : B \rightarrow p)
⊢ \lambda_{t}.for \cdot t; Acc \rightarrow For; \lambda_{x}.x
⊢ \lambda_{st}.easy \cdot s \cdot t; For \rightarrow (PRO \rightarrow Inf) \rightarrow It \rightarrow PrdA; \lambda_{xPo}.easy x P
⊢ \lambda_{sf}.easy \cdot s \cdot (f e); For \rightarrow (Acc \rightarrow PRO \rightarrow Inf) \rightarrow PRO \rightarrow PrdA;
\lambda_{xry}.easy x (r y)
How Neutral Expressions Get Case

\[ \vdash \lambda s.t. \cdot \text{pleases} \cdot t; \text{Nom} \rightarrow \text{Acc} \rightarrow S \]

\[ \vdash \lambda t.john \cdot \text{pleases} \cdot t; \text{Acc} \rightarrow S; \text{please } j \]

\[ \vdash \text{john} \cdot \text{pleases} \cdot \text{mary}; S; \text{please } j \text{ m} \]
Nonpredicative “Prepositional” Phrases

Here and henceforth, leaves with overbars were already proved as lemmas in earlier derivations.

\[
\begin{align*}
\vdash \lambda_t. \text{for} \cdot t; \text{Acc} &\rightarrow \text{For}; \\
\lambda_x.x &\vdash \text{mary}; \text{Acc}; m \\
\vdash \text{for} \cdot \text{mary}; \text{For}; m
\end{align*}
\]

In the absence of clear empirical support for calling nonpredicative For-phrases ‘prepositional’, we just treat For as a basic tecto.
A Base Phrase

\[ \vdash \lambda_t. \text{please} \cdot t; \text{Acc} \rightarrow \text{PRO} \rightarrow \text{Bse}; \lambda_{yx}. \text{please} \ x \ y \quad \vdash \text{john}; \text{Acc}; j \]

\[ \vdash \text{please} \cdot \text{john}; \text{PRO} \rightarrow \text{Bse}; \lambda_x. \text{please} \ x \ j \]
An Infinitive Phrase

[1:]

\[ \vdash \lambda t. \text{to } \cdot t; (A \rightarrow \text{Bse}) \rightarrow A \rightarrow \text{Inf}; \lambda P.P \quad \vdash \text{please } \cdot \text{john}; \text{PRO } \rightarrow \text{Bse}; \lambda x.\text{please } x \ j \]

\[ \vdash \text{to } \cdot \text{please } \cdot \text{john}; \text{PRO } \rightarrow \text{Inf}; \lambda x.\text{please } x \ j \]

- Here \( A \) (two occurrences) in the \textit{to} schema was instantiated as PRO (and \( B \) in \( P : B \rightarrow p \) as e). This is legitimate because the schematization is over \( A \leq \text{NP} \), and in fact \( \text{PRO} \leq \text{NP} \).

- This is an instance of (the LG counterpart of) Raising, in this case of PRO from the base-form complement \text{please John} to the infinite phrase.

- There is no \textit{sign} of tectotype PRO that ‘raises’!
An Impersonal Predicative Phrase

[2]:

\[ \vdash \lambda_{st}. \text{easy} \cdot s \cdot t; \text{For} \rightarrow (\text{PRO} \rightarrow \text{Inf}) \rightarrow \text{It} \rightarrow \text{PrdA}; \lambda_{xPo}. \text{easy} \cdot x \cdot P \]

\[ \vdash \lambda_{t}. \text{easy} \cdot \text{for} \cdot \text{mary} \cdot t; (\text{PRO} \rightarrow \text{Inf}) \rightarrow \text{It} \rightarrow \text{PrdA}; \lambda_{Po}. \text{easy} \cdot m \]

[3]:

\[ \vdash \text{easy} \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please} \cdot \text{john}; \text{It} \rightarrow \text{PrdA}; \lambda_{o}. \text{easy} \cdot m \cdot (\lambda_{x}. \text{please} \cdot x \cdot j) \]

\[ \vdash \text{easy} \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please} \cdot \text{john}; \text{It} \rightarrow \text{Prd}; \lambda_{o}. \text{easy} \cdot m \cdot (\lambda_{x}. \text{please} \cdot x \cdot j) \]

- This is just like a Control construction, e.g. *Mary tries to please John*, which means *try m (\lambda_{x}. \text{please} \cdot x \cdot j)* ...

- Except that the controller is the For-phrase, rather than the subject (which is only a dummy)

- This uses the semantics for Control where the infinitive complement is analyzed as a property rather than a proposition.
It is easy for Mary to please John

[4]:

\[
\begin{align*}
\vdash & \lambda_{st}.s \cdot is \cdot t; A \rightarrow (A \rightarrow Prd) \rightarrow S; \lambda_{xP}.P \ x & \vdash it; It; \\
\vdash & \lambda_{t}.it \cdot is \cdot t; (It \rightarrow Prd) \rightarrow S; \lambda_{P}.P \ * \\
\vdash & it \cdot is \cdot easy \cdot for \cdot mary \cdot to \cdot please \cdot john; S; easy \ m \ (\lambda_{x}.please \ x \ j)
\end{align*}
\]

■ Here \( A \) in \( is \) is instantiated as \( It \) (and \( B \) in \( x : B \) as \( T \), so \( P : T \rightarrow p \)).

■ This is another case of ‘Raising’, in this case of the unrealized \( It \) subject of \textit{easy for Mary to please John}.

■ In no sense was the sign \textit{it} ever in the predicative phrase.
A Gappy Infinitive Phrase

[5]:

\[ \vdash \lambda_t. \text{please } t; \text{Acc} \rightarrow \text{PRO} \rightarrow \text{Bse}; \lambda_{yx}. \text{please } x \ y \quad s; \text{Acc}; y \vdash s; \text{Acc}; y \]

\[ \quad \vdash s; \text{Acc}; y \vdash \text{please } s; \text{PRO} \rightarrow \text{Bse}; \lambda_x. \text{please } x \ y \]

[6]:

\[ \vdash \lambda_t. \text{to } t; (A \rightarrow \text{Bse}) \rightarrow A \rightarrow \text{Inf}; \lambda_P.P \quad [5] \]

\[ \quad \vdash s; \text{Acc}; y \vdash \text{to } \text{please } s; \text{PRO} \rightarrow \text{Inf}; \lambda_x. \text{please } x \ y \]

\[ \vdash \lambda_s. \text{to } \text{please } s; \text{Acc} \rightarrow \text{PRO} \rightarrow \text{Inf}; \lambda_{yx}. \text{please } x \ y \quad \text{HP} \]

- The object trace, which is withdrawn in the last proof step, captures the sense in which ‘Tough-Movement’ works like an $\overline{A}$ (long-distance) dependency.

- The $\lambda_s$ and $\lambda_y$ in the pheno and semantics of the conclusion are prefigured by the empty operator binding the trace in Chomsky’s (1977) analysis of this same construction:

  \[ [\text{john}_i \text{ is easy } \text{O}_i[\text{PRO to please t}_i]] \]

- Unlike Hicks’ analysis, there is nothing ‘complex’ about the operator that binds the trace (it is just $\lambda$), and no sense in which anything ever ‘moves out’ of it.
A Personal Predicative Phrase

[7:]

\[ \vdash \lambda s f . easy \cdot s \cdot (f e); \text{For} \rightarrow (\text{Acc} \rightarrow \text{InfP}) \rightarrow \text{PRO} \rightarrow \text{PrdA}; \lambda x R y . easy \ x \ (R y) \quad \vdash \text{for} \cdot \text{mary}; \text{For}; \text{m} \]

\[ \vdash \lambda f . easy \cdot \text{for} \cdot \text{mary} \cdot (f e); (\text{Acc} \rightarrow \text{InfP}) \rightarrow \text{PRO} \rightarrow \text{PrdA}; \lambda R y . easy \ m \ (R y) \]

[8:]

\[ \vdash easy \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please}; \text{PRO} \rightarrow \text{PrdA}; \lambda y . easy \ m \ (\lambda x . \text{please} \ x \ y) \quad \text{D2} \]

\[ \vdash easy \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please}; \text{Nom} \rightarrow \text{PrdA}; \lambda y . easy \ m \ (\lambda x . \text{please} \ x \ y) \quad \text{D3} \]

\[ \vdash easy \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please}; \text{Nom} \rightarrow \text{Prd}; \lambda y . easy \ m \ (\lambda x . \text{please} \ x \ y) \]

- Here ‘InfP’ abbreviates PRO → Inf.
John is easy for Mary to please

[9:]

\[ \vdash \lambda s.t. \cdot s \cdot t; A \rightarrow (A \rightarrow \text{Prd}) \rightarrow S; \lambda x.P \cdot x \quad \vdash \text{John}; \text{Nom}; j \]
\[ \vdash \lambda t.\text{john} \cdot \text{is} \cdot t; (\text{Nom} \rightarrow \text{Prd}) \rightarrow S; \lambda P.P \cdot j \]

\[ [9] \quad \vdash \text{easy} \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please}; \text{Nom} \rightarrow \text{Prd}; \lambda y.\text{easy} \cdot m \cdot (\lambda x.\text{please} \cdot x \cdot y) \]
\[ \vdash \text{John} \cdot \text{is} \cdot \text{easy} \cdot \text{for} \cdot \text{mary} \cdot \text{to} \cdot \text{please}; S; \text{easy} \cdot m \cdot (\lambda x.\text{please} \cdot x \cdot j) \]

- Here \( A \) in \( is \) is instantiated as \( \text{Nom} \).
- This is another instance of ‘Raising’, in this case of the unrealized \( \text{Nom} \) subject of the predicative phrase \( \text{easy for Mary to please} \) to the sentence.
- But \( \text{John} \) was never actually in the predicative phrase!
- \( \text{Tough} \)-constructions are unproblematic for \( \text{LG} \).