1 Review of Government and Binding Theory

Brief Overview of Government and Binding Theory

- **Syntactic Derivation**
  1. X-Theory PSRs create Deep Structure (DS)
  2. Transformational Rules yield Surface Structure (SS)
  3. Covert Movement Rules yield Logical Form (LF)

- **Theta-roles and Case are assigned under government**

- **Binding Theory**: Structural restrictions on DPs and coreference

  Identifying the level at which binding principles hold = notoriously complicated!

- **Logical Form**:
  1. Quantifier Raising
  2. Covert Wh-Movement
  3. Reconstruction

  This movement (originally) was not to account for the surface position of NPs, but rather their interpretation

  Movement from A-Positions to A'-Positions

- **The single movement operation, Move-α is unconstrained**

- **Applications of Move-α leave behind structural elements, however, i.e., traces and chains**

- **Restrictions and filters rule out structures that Move-α’s freedom of application overgenerates**

- **A lot of these restrictions/filters have the same theme: Don’t bother with extraneous movement! Don’t move too far!**

- **Question**: Do we really need all this stuff? eg., the four levels of representation, the many filters?

2 Paradigm Shift: From GB to Minimalism

→ The Minimalist Program is Economy-Driven

1. Methodological Economy (Hornstein 2000)
   i.e., Ockham’s Razor, the simpler and fewer things, the better.

2. Substantive Economy (Hornstein 2000)
   Least effort (eg., movements)

- **Chomsky presents Minimalism as a program, not a theory**
  i.e., revising the kinds of questions and theories linguists ask and propose

Minimalist Goals: Theoretical Elegance!

1. Minimize levels of representation (eg., DS, SS, LF, PF...)
   → Shift towards an explicitly derivational approach

2. Minimize theoretical constructs (eg., pro, traces, categories, ...)

3. Minimize structure (eg., non-branching bar-levels)

4. Minimize operations (eg., fewer, and shorter)

5. ...
3 The Minimalist Program (based on Adger 2003)

3.1 The Lexicon and Lexical Features

\[ G = (L, O) \]
\[ L = \text{Lexicon (Set of LIs)} \quad O = \text{Set of operations} \]

- A syntactic derivation selects a set of lexical items, called the numeration eg., \{Harry, receive, a, letter, from, Hogwarts\}
- Lexical items have features
  1. Semantic features
  2. Phonetic Features
  3. Syntactic features
    (i) Category features
    (ii) Inflectional features
    (iii) Selectional features

- Features are what trigger the operations merge and move eg., from has Category feature: P
  Selectional features: uD

- Uninterpretable features, \([uF]\), must be deleted by LF!

**Full Interpretation:** The structure to which the semantic interface rules apply (LF) contains no uninterpretable features.

- Q: How do you get rid of uninterpretable features? By checking them!

**The Checking Requirement:** Uninterpretable (c-selectional) features must be checked; once checked, they can delete.

- Q: How do you check features? Sisterhood with F!

**Checking Under Sisterhood:** An uninterpretable c-selectional feature F on a syntactic object Y is checked when Y is sister to another syntactic object Z which bears a matching feature F

- Q: How do you form a sisterhood relation? Merging and moving!

3.2 A Basic Minimalist Derivation (Chomsky 1995)

**Syntactic Operations, O**

1. select, 2. merge, 3. move (We’ll discuss Chomsky 2001 next week)

- Select copies lexical items from L, and puts these into the numeration; the derivation can only use elements in the numeration

**The Inclusiveness Condition:** The LF object \(\lambda\) must be built only from features of the lexical items of the numeration. (Hornstein et al. 2005)

- Merge combines two elements from the workspace, \(\alpha, \beta\), and creates a new object, \(\gamma = \{\alpha, \beta\}\) in the workspace

**Numeration:** \(\{\alpha, \beta, \delta\}\)

\[ \text{Merge}(\alpha, \beta) \rightarrow \gamma \]

\[ \alpha \beta \]

Recall: Minimize operations! Not 5 X’-PSRs, but 1 operation

- Move: Applies to a phrase marker, \(\beta\), with constituents \(\alpha\) and \(\delta\), Move copies \(\alpha\) and Merges \(\alpha\) with \(\beta\) to form the new object \(\gamma\)

\[ \text{Move} \]

\[ \beta \]

\[ \alpha \delta \]

\[ \alpha \beta \]

\[ \gamma \]

- Move must be triggered by the need to check an uninterpretable feature; Feature Checking is notated with a strike-through:

\[ \text{Merge is strictly binary!} \]
Merge(from[[P, uD]],

Hogwarts [[D]])

→

If you want to check the [uP] feature on receive, you need to check the [uD] feature on the P first!

✓

VP, [V]

V', [V, uD]

DP, [D]

V0, [V, uP, uD]

PP, [P]

a letter

receive

from Hogwarts

Structural Definitions

Maximal Projections (X^max/XP): Syntactic objects with all of their selectional features checked.

Intermediate Projections (X'): Syntactic objects (that are not lexical items) that still have selectional features to check

Minimal Projections X^min/X^0: Lexical items selected from the numeration.

→ Given these definitions, X^max and X^min are not mutually exclusive

→ A lexical item with no uninterpretable features is both a X^min and a X^max

→ Intermediate projections cannot be minimal or maximal projections

→ This sort of relational system avoids vacuous/non-branching intermediate projections (i.e., all those annoying X’s)

Recall: Minimize structure! Only X’s where you need them!

→ An intermediate projection only occurs if there is a second [uF] that needs check (e.g., by merging a specifier):

- The **Head** is defined as the syntactic item which selects in the Merge operation (i.e., *from* in the case above)

- **Headedness** is the property of heads projecting their syntactic features to the syntactic object that Merge creates:

✓

[V]

[V, uD]

receive

from Hogwarts

×

[D]

[P]

from Hogwarts

• **Merge** can only apply to the (i) lexical items in the numeration and (ii) the root of constructed syntactic objects (i.e., the topmost node)

This is referred to as:

**The Extension Condition**: A syntactic derivation can only be continued by applying operations to the root of the tree.

• This means any uninterpretable features on α and β must be on the head! Otherwise they won’t be projected, and cannot be checked, which will cause the derivation to **Crash** as opposed to **Converge**!
**Two Arguments:**

\[ \text{Max} \]

\[ X \]

\[ \text{COMP} \]

\[ \text{SPEC} \]

\[ X' \]

\[ \text{Max} \]

\[ X \]

\[ \text{COMP} \]

\[ X_0 \]

**One Argument:**

\[ \text{Max} \]

\[ X \]

\[ \text{COMP} \]

\[ X_0 \]

**No Arguments:**

\[ X_0 \]

\[ / \]

\[ \text{Max} \]

**GB/Extended X'-Theory:** Primitives

1. Three category-neutral PSRs (+ conjunction)
2. Three levels of structure (X, X',XP)
3. Three phrasal positions (complement, adjunct, specifier)

**Minimalism**

1. One combinatorial operation: **MERGE**
   
   → Levels of structure and phrasal positions are derived from the uninterpretable features of the lexical items involved

**Practice Deriving Minimalist Trees:**

Assume LIs with morphosyntactic features as follows:

- **send** = \{V, uD, uP, ACC\}
- **eat** = \{V, uD, ACC\}
- **cry** = \{V, ACC\}
- **to** = \{P, uD, OBL\}
- **with** = \{P, uD, OBL\}
- **on** = \{P, uD, OBL\}
- **the** = \{D, uN, uACC\}
- **a** = \{D, uN, uOBL\}
- **this** = \{D, uN, uACC\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}
- **∅** = \{D, uN, uPROP, uOBL\}

1. **SELECT** items for your **NUMERATION**
2. **MERGE** and **MOVE** LIs to **CHECK UNINTERPRETABLE FEATURES**
3. Does your derivation **CRASH** or **CONVERGE**?

   (a) Harry sent the letter on the table to Hermione
   
   (b) Hermione ate a sandwich with Ron
   
   (c) Ron cries in the bathroom with a sandwich
   
   (d) (make your own sentence)

**4 BPS - Bare Phrase Structure (Hornstein et al. 2005)**

Recall: **Minimize theoretical constructs!**

- The information that the symbols \text{max}, \text{'} , and \text{0} encode can be read from the nodes’ relations to other projections in the tree...

...so if you want to be really minimalist, get rid of them by using **Bare Phrase Structure** (BPS)!

- BPS also removes lexical categories from the syntax - i.e., you don’t label nodes with lexical categories like N, N', NP, V, V', VP

  → If lexical category is already represented in the lexicon, it needn’t be replicated in the syntax

  Having the same information encoded TWICE in the grammar is redundant, not minimal

- When using **Bare Phrase Structure**, the syntactic object created from \( α \) and \( β \) by **MERGE** is labelled as follows, where \( α \) is the head:

  \{α, [α, β]\}

- This yields structures with labels as follows:
5 The T-Model VS Y-Model

### The GB T-Model

**D-structure**
- Move-α

**S-structure**
- Move-α

**PF** → **LF**

#### Recall:

- Minimize theoretical constructs!
- Including levels of representation (DS, SS, LF, PF)

#### The Minimalist Y-Model

**Numeration** = \{LI_1, LI_2, LI_3, ...\}

- **Move-α**

<table>
<thead>
<tr>
<th>PF</th>
<th>LF</th>
</tr>
</thead>
</table>

**Sentence:**

- A minimalist approach maintains PF and LF
  - PF is obligatory
    - sentences clearly have a phonetic/phonological component
  - LF is obligatory
    - sentences clearly have a semantic component

- But do we really need DS and SS?

- Q: Don’t we need all these levels of representation?
  - If we don’t have DS, how do we satisfy the **Theta-criterion**?
  - If we don’t have SS, how do we satisfy the **Case Filter**?\(^2\)

- **Case** is just **Uninterpretable Features** that are checked during the course of the derivation

- **Crashing** will occur if there are unchecked UFs at LF

---

\(^2\)There were many more theoretically-motivated and empirically-motivated reasons to have SS, but we didn’t cover these (they would have required a much deeper look into GB). See Hornstein et al. 2005 for more these motivations, and how a minimalist approach can account for them.
5.1 Raising VS Control - Evidence for DS?

• Empirical Differences (Hornstein et al. 2005 - 2.3.2.2)
  1. Subject Theta-role Assignment
  2. Subject Expletives
  3. Phrasal Idioms (see (Hornstein et al. 2005)
  4. Active-Passive Entailments

1. Control predicates assign the subject a theta-role; raising predicates do not

(1) a. Nat hoped to arrive on time.  
   hopeV: θ_EXP, θ_CONTENT
   Control

b. Nat seemed to arrive on time. 
   seemV: θ_SITUATION
   Raising

(2) a. DS: Nati hoped PROi to arrive on time. 
   Control

b. DS: ____ seemed Nat to arrive on time. 
   Raising

• The distinct theta-roles that the subject bears was explained as a difference in DS position (where theta-roles are assigned)
• Thus the contrast between (1a) and (1b) can be seen as empirical evidence for the existence of DS
• But there’s no reason why θ-role assignment must occur at a particular level of representation (i.e., a ‘where’)
• θ-roles assignment can have a derivational (i.e., when) restriction instead:

(3) Theta-Role Assignment Principle (Hornstein et al. 2005)
θ-roles can only be assigned under a Merge operation

2. Raising predicates can have expletive subjects; control predicates cannot

(4) a. *It hopes that Tee arrives late 
   hopeV: θ_EXP ×, θ_CONTENT ✓

b. It seems that Tee arrives late 
   seemV: θ_SITUATION ✓

• Assumption: Expletives cannot bear θ-role
• (4a) is ungrammatical because hopeV needs to assign its θ_EXP theta-role at DS, and there is no appropriate DP
• But, assuming TRAP, we don’t need a particular representational level (i.e., DS) where the theta-criterion is assessed
  – θ-role selection is formalized via uFs, which must be deleted by LF
  – These particular uFs can only be deleted on Merge (acc. to TRAP)

4. Raising maintains the entailment relation between passive-active; control structures do not

(5) John seemed to be examined by the doctor 
   Raising ≈ The doctor seemed to examine John

(6) a. DS: the doctor iri hoped PROi to examine John 
   Control

b. DS: i seemed [to be examined John by the doctor] 
   PASSIVE

(7) John hoped to be examined by the doctor
   CONTROL ≠ The doctor hoped to examine John

(8) a. DS: The doctori hoped [PROi to examine John] 
   Control

b. DS: John hoped [to be examined PROi by the doctor] 
   PASSIVE

• The control verb hopeV assigns its θ_EXP to different DPs in the DS of the active VS the DS of the passive
• In a minimalist approach, the same info about $\theta$-roles is visible at LF

• The initial trace shows where the XP first merged\(^3\)

(9) John seemed to be examined by the doctor
$\approx$ The doctor seemed to examine John

a. LF: [the doctor$\text{i}$ seemed $\text{v}$ to examine John$\text{t}$]

b. LF: [John$\text{i}$ seemed $\text{v}$ to be examined $\text{v}$ by the doctor$\text{t}$]

(10) John hoped to be examined by the doctor
$\#$ The doctor hoped to examine John

a. LF: [[The doctor hoped [PRO to examine John$\text{v}$]]

b. LF: [[[John hoped [PRO$\text{i}$ to be examined $\text{v}$ by the doctor$\text{t}$]]]

• **Conclusion:** DS is actually not required to account for the difference between control/raising constructions

• **Next Week:** Minimalism II
  1. Consequences of Binary Merge: Causative little $\text{v}$
  2. The Copy-Merge theory of Movement
  3. “Covert Movement” in Minimalism
     - Agree under C-Command
     - Move-F (strong vs weak features)

**References**


\(^3\)Assuming, as we are at this point, that traces exist.