MINIMALIST PROGRAM READING GROUP

From Features to Structures

Architecture

- UG makes available a set **F** of features (linguistic properties) and operations $\mathbf{C_{HL}}$ (the computational procedure for human language) that access F to generate expressions **Exp.** (*MI* 100)
- The language L maps F to a particular set of expressions.
- L makes a one-time selection of a subset [F] (or $[F_L]$ in *DP* 7) of F, dispensing with further access to F. (*MI* 100) These are the features that enter into L: others can be disregarded in use of L. (*DP* 7)
- L includes a one-time operation that assembles elements of [F] into a lexicon **Lex**, with no new assembly as computation proceeds. (*MI* 100) L assembles $[F_L]$ to lexical items **LI** of a lexicon Lex, the LIs then entering into computations as units. The entry LI is a once-and-for-all collection of phonological, semantic, and formal features. (*DP* 7-8)
- Narrow syntax maps a selection of choices from LEX to LF; the phonological component has further access to $[F_L]$. (*DP* 8)
- Derivations make a one-time selection of a lexical array **LA** from Lex,[1] then map LA to expressions, dispensing with further access to Lex. (*MI* 100)
- Acquiring a language: selection of the features [F], construction of lexical items Lex, and refinement of C_{HL} in one of the possible ways- parameter setting. (*MI* 100)
- A language L maps ([F], Lex) to Exp. However, in **narrow syntax** (the computation of LF), C_{HL} is a mapping of Lex to the LF representation of Exp.

$$F \Rightarrow [F] \Rightarrow Lex \Rightarrow LA \Rightarrow Exp$$

- Language L follows the following procedures (*MI* 101)

[1] A 'numeration' **N** if we distinguish independent selections of a single lexical item (*MI* fn28). C_{HL} is mapping some array A of lexical choices to the pair (π, λ). A must indicate what the lexical choices are and how many times each is selected by C_{HL} in forming (π, λ). A numeration is a set of pairs (LI, *i*), where LI is an item of the lexicon and *i* is its index, understood to be the number of times that LI is selected. Take A to be (at least) a numeration N; C_{HL} maps N to (π, λ). The procedure C_{HL} selects an item from N and reduces its index by 1, then performing permissible computations. A computation constructed by C_{HL} does not count as a derivation at all, let alone a convergent one, unless all indices are reduced to zero. (*MP* 225)

☞ to specify the language: Select [F] from the universal feature set F.
Select Lex, assembling features from [F].

☞ to derive a particular Exp: Select LA from Lex.
Map LA to Exp, with no recourse to [F] for narrow syntax.

Operations

Merge

- **Merge** takes two syntactic objects (α,β) and forms K (α,β) from them. (*MI* 101)

  Merge takes a pair of syntactic objects (SO_i, SO_j) and replaces them by a new combined syntactic object SO_{ij}. (*MP* 226)

  Applied to two objects α and β, Merge forms the new object K, eliminating α and β. K must be constituted somehow from two items α and β. The simplest object constructed from α and β is the set {α,β}, so we take K to involve at least this set, where α and β are the *constituents* of K. K must be of the form {γ ,{α,β}, where γ identifies the type to which K belongs, indicating its relevant properties. γ is the *label* of K. (*MP* 243)

  γ is either α or β; one or the other *projects* and is the *head* of K. If α projects, then K = {α, {α,β}}. (*MP* 244)

  The label is redundant. The syntactic objects are LIs, or sets {α, β} or <α, β> constructed from them. The label is determined and available for operation within C_{HL} or for interpretation at the interface. (*MI* 135)

- Merge 'comes free'. (*DP* 2)
- When α, β merge, it is to satisfy (selectional) requirements of one (the **selector**) but not both. (*MI* 133)
- In K(α,β), Merge provides two natural relations (*MI* 116, *DP* 2):

  Sisterhood/Sister: holds of (α, β)

  The sisterhood relation is significantly primarily for heads, i.e. LIs and modified LIs (MLIs) formed from them.[2]

  Sisterhood relations remain if LI is modified to MLI. LI and its modifications are not distinguished with regard to the fundamental relations defined in terms of Merge.

  Immediately-Contain (IC): holds of (K, α), (K, β), and (K, K)

- Three new relations are derived (*MI* 116):

  Contain: K contains α if K immediately contains α or immediately contains L that contains α; α is a *term* of K if K contains α.

[2] An MLI is an LI with uninterpretable features deleted. (*MI* 126)

Identity: = (sister(sister))

C-command: = (sister(contain)), α c-commands β if α is the sister of K that contains β.

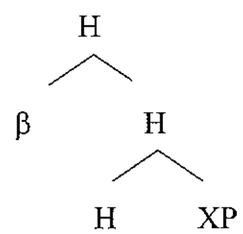
The relation of c-command: only in interpretation of the information it provides? I.e. in mapping it to syntactic objects that belong to mental systems external to the language faculty itself. (*MI* 116)

- The new object K formed by Merge of β to α retains the label L of α, which projects. There are two possibilities: (*MI* 136)

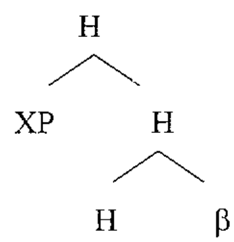
  Extension Condition: α is unchanged.

  Local Merge: β is as close to L as possible

- (a) $\{\beta, \{XP, H\}\}$



  (b) $\{XP, \{\beta, H\}\}$



- Given a choice of operations applying to α and projecting its label L, select one that preserves R(L, γ). (*MI* 137)

<u>Agree</u>

- A **probe** P seeks a **goal** G: 'matching' features that establish agreement. (*MI* 122) We can take **Match** to be Identity. Matching of probe-goal induces Agree, eliminating uninterpretable features that activate them. (*DP* 4)
  - (a) Probe and goal must both be active for Agree to apply.
  - (b) α must have a complete set of ϕ-features (it must be ϕ-*complete*) to delete uninterpretable features of the paired matching element β.
- **Agree** establishes a relation (agreement, Case checking) between an LI α and a feature F in some restricted search space (its *domain*). Agree is language-specific, never built into special-purpose symbolic systems and apparently without

significant analogue elsewhere. It relates to the design conditions for human language. (*MI* 101)

(a) T be elected an unpopular candidate

(there are 3 kinds of uninterpretable features: ϕ-features of T, the EPP-feature of T, and the structural Case feature of *an unpopular candidate*)

probe: the ϕ-set of T

goal: the ϕ-set of *candidate*

the relation of the probe of T to its goal: *T-associate relation*

(b) there T-was elected an unpopular candidate

(there is long-distance agreement of T and its goal/its associate.)

- Locating the goal, the probe erases under matching.
- Taking structural Case to be a reflex of an uninterpretable ϕ-set, it too erases under matching with the probe. (*MI* 122) Structural Case is not a feature of the probes (T, *v*), but it deletes under agreement if the probe is appropriate: ϕ-complete. Case itself is not matched, but deletes under matching of ϕ-features. (*DP* 4)
- The EPP-feature of T must be satisfied by 'pied-piping' of a phrase P(G).
- Agree: the erasure of uninterpretable features of probe and goal. (*MI* 122)
- The agreement relation removes the uninterpretable features from narrow syntax, allowing derivations to converge at LF while remaining intact for the phonological component (with language-variant PF-manifestation). (*DP* 2-3)
- G must (at least) be in the *domain* D(P) of P and satisfy locality conditions. (*MI* 122)
  - (a) Matching is feature identity.
  - (b) D(P) is the sister of P.
  - (c) Locality reduces to 'closest c-command'.
- 'Feature identity' is identity of the choice of feature, not of value. (*MI* 124)
- D(P) is the c-command domain of P. A matching feature G is *closest to* P if there is no G' in D(P) matching P such that G is in D(G'). (*MI* 122)
- **Equidistance**: Terms of the same minimal domain are 'equidistant' to probes. (*MI* 122)
- The **minimal domain** of a head H is the set of terms immediately contained in projections of H. (*MI* 123)
- 'Checking' reduces to deletion under matching with an active local goal and ancillary deletion of the uninterpretable feature that rendered the goal active. (*MI* 123)
- General conclusions: (*MI* 126)
  - (a) Long-distance agreement is a T-associate (probe-goal) relation.

(b) The EPP can be satisfied by

    (i) Merge of expletive

    (ii) Merge of associate

    (iii) Merge of α closer to T than the associate.

- Merge of (α, β) has some of the properties of Agree: a feature F of one of the merged element must be satisfied for the operation to take place. (*MI* 134)
- The selector F for Merge is analogous to the probe for Agree. (*MI* 134)
- The operations Merge and Agree must: (*MI* 132)
  - (a) Find syntactic objects to which they apply.
  - (b) Find a feature F that derives the operation.
  - (c) Perform the operation, constructing a new object K.

Move

- **Move** combines Merge and Agree. Move establishes agreement between α and F and merges P(F) to αP where P(F) is a phrase determined by F (perhaps but not necessarily its maximal projection) and αP is a projection headed by α. P(F) becomes the specifier of α ([Spec, α]). (*MI* 101)
- If α in the syntactic object SO is merged somewhere else (by Move) to form SO', the two occurrences of α constitute a *chain*. (*MI* 114)

  Chain: a sequence of occurrences of a single α.

  An **occurrence** of α is a sister of α.
  - (a) I expect a proof to be discovered.
  - (b) I_i T [t_i expect [a proof_j to be discovered t_j]]

    Two chains: $C_I = \langle I_1, I_2 \rangle$ and $C_P = \langle P_1, P_2 \rangle$

    I_1 = *T ... discovered* t_j

    I_2 = *expect ... discovered* t_j

    P_1 = *to be discovered* t_j

    P_2 = *discovered*
- Chains: sequences → sets
  - (a) who did you say [*t* has [*t'* discovered the proof]]

    A'-chain: (*who, t*)

    A-chain: (*t, t'*)

    3 occurrences of *who*: {Q_1, Q_2} and {R_1, R_2}

      Q_1 = *did you say ... proof*

      Q_2 = R_1 = *has ... the proof*

      R_2 = *discovered the proof*
- A chain is a set of occurrences of an object α in a constructed syntactic object K. (*MI* 116)

- What guarantees that the feature is deleted throughout the chain? The feature is deleted in the single element α.
- Problem 1: An occurrence of K is an X' category, neither X^0 nor X^{max}, hence arguably invisible. (*DP* 32)
  - ⇒ the conceptual and empirical arguments for X'-invisibility are slight.
  - ⇒ define 'occurrence' in terms of Immediately-Contain instead of Sister.
- Problem 2: A chain is not a 'syntactic object'. (*DP* 32)
  - ⇒ no operations of L apply to chains.

- **Merge over Move**: Move is a more complex operation as it involves extra steps, e.g. 'pied-piping'.

  [I]

  (a) to be someone in the room

  [II]

  (b) <u>there</u> to be someone in the room (by Merge)

  (c) <u>someone</u> to be *t* in the room (by Move)

  [III]

  (d) There seems *t* to be someone in the room.

  (e) *There seems someone to be in the room.
- Four kinds of complexity considerations: (*MI* 105)
  a. Simple operations preempt more complex ones.
  b. Search space is limited (locality).
  c. Access to the feature set F is restricted by the procedures in (3).
  d. Computation is locally determined (no look-ahead).

Features

- There are uninterpretable features of lexical items. (*MI* 119)

  E.g. structural Case

  E.g. agreement features: semantically interpretable for nouns, but not for verbs/adjectives, and phonetically optional throughout.
- Manifestation of structural Case depends on interpretable features of the probe: finite T (nominative), *v* (accusative), control T (null) (*MI* 123).

  ∴ structural Case is a 'single undifferentiated' feature (*MI* 124)

Categories

- There are two types of categories: **substantive** and **functional**. (*MI* 102)
- Core functional categories **CFCs** are C (force/mood), T (tense/event structure), and *v* (the 'light verb' head of transitive constructions).
  The D head of DP seems to belong to a different system. (*MI* fn 142)
- All CFCs may have ϕ-features (obligatory for T and *v*). ϕ-features of CFCs are uninterpretable, constituting the core of the systems of (structural Case) assignment and 'dislocation' (Move).
- **S-selection** of CFCs:
  (a) C can be unselected (root). It is selected by substantive categories.
  (b) *v* only by a functional category.
  (c) T is selected by C or V. If selected by C. it has a full complement of ϕ-features; if by V, it is *defective* (T_{def}).
  (d) T and *v* select verbal elements. *v* may select a nominal phrase NP/DP as its external argument.
- Extra Spec of CFCs, **EPP-features**
  (a) C: a raised *wh*-phrase
  (b) T: the surface subject
  (c) *v*: the phrase raised by object shift
- EPP-features are uninterpretable though the configuration they establish has effects for interpretation.
- Basic structural properties of CFCs (*MI* 102-103)

  α = [XP [(EA) H YP]]

  a. If H is *v*/C, XP is not introduced by pure Merge.[3]
  b. In the configuration [$_\beta$ T_β…α], β minimal,
     i. if H is C, T_β is independent of α;
     ii. if H is *v*, T_β agrees with EA, which may raise to [Spec, T_β] though XP cannot;
     iii. if H is T_{def}, XP raises to [Spec, T_β] if there is no closer candidate γ for raising.

  ☞ see examples (7-9) in *MI* 103 and a revised version in *MI* 129
- Pure Merge in θ-position is required of (and restricted to) arguments. (*MI* 103)
- The θ-theoretic principle bars pure Merge of arguments in non-θ-positions and restricts Move to such positions. (*MI* 106).

[3] **Pure Merge** is Merge that is not part of Move. (*MI* 103)

Topics for the next meeting

(7/19/2000, 2pm, QT502, PolyU)

- Phases
- Movement and cyclicity

  Reading materials: Chomsky 2000:§3 and 6, Chomsky 1999

References

Chomsky, Noam. 1995. Categories and transformation. In *The Minimalist Program.* 219-394. Cambridge, Mass.: The MIT Press. (***MP***)

Chomsky, Noam. 1999. Derivation by phase. *MIT Occasional Papers in Linguistics 18.* Cambridge, Mass.: MITWPL. (***DP***)

Chomsky, Noam. 2000. Minimalist inquiries: the framework. In *Step by step: essays on minimalist syntax in honor of Howard Lasnik*, eds., Roger Martin, David Michaels, and Juan Uriagereka, 89-155. Cambridge, Mass.: The MIT Press. (***MI***)

(prepared by Sze-Wing Tang)