Computing Minimalism

Sandiway Fong
Departments of Linguistics and Computer Science
University of Arizona

this work is in part jointly developed with Jason Ginsburg, U. of Aizu, Japan

Minimal Computation (MC)

Chomsky (2011)

• language is a computational system:
  – Generative Procedure (GP)
  – Parsing Procedure (PP)
• we should consider computational efficiency, but...
• when it comes to Minimal Computation (MC):
  – conflicts between GP and PP are always resolved in favor of GP

(12) $S' \{\{\text{that book}\}, S\} = \{\{\text{that book}\}, \{\text{John, \{read, \{that, book\}\}}\}\}$

In (12) there are two copies of the object $\{\text{that, book}\}$. $S'$, with the two copies, has the right form for CI (conceptual-intentional) but not of course for SM (sensory-motor)—first because linearization is required and second because the hierarchically less prominent copy is not pronounced. The latter property follows at once from MC: at least the most prominent copy must be pronounced, or there will be no evidence that topicalization took place at all, but computation is reduced if at most one copy is pronounced—massively reduced in nontrivial cases.
Minimal Computation (MC)

- Chomsky (2011)

also follows from MC. The inadequacy is severe. Anyone who has worked on parsing programs knows that a major difficulty is posed by “filler-gap” problems: given who in (10), the problem for parsing/perception is to locate the gap where it receives its interpretation in the argument structure of the sentence, not a trivial matter in general. These problems would largely be obviated if all copies were pronounced, violating MC.

In brief, where there is a conflict between communicative and computational efficiency, the latter seems to win, hands down. It appears that Aristotle’s dictum should be reversed: language is not sound with meaning but meaning with sound, a very different matter. Externalization by the SM system appears to be a secondary property of language.

.. explore idea that the Generative Procedure (GP) and Parsing Procedure (PP) share architecture

*PP can help GP minimize computation*
What constitutes an implementation?

• Minimalism:
  – (Chomsky 2001) and later papers

• Components of the theory:
  1. Recursive Merge (internal/external)
  2. Probe/Goal search (active/inactive)
  3. Agreement (value Case, interpretable/uninterpretable features)
  4. Phases (limits on search)
What constitutes an implementation?

assume, are raising constructions and their exceptional-Case-marking (ECM) counterparts, as shown schematically in (4a), where $\beta$ is the matrix clause, $\alpha$ is an infinitival with YP a verbal phrase (the case most relevant here), and $P$ is the probe: $T$ with a raising verb (case (4b)), $v$ with an ECM transitive verb (case (4c)).

(4) a. $[\alpha P [\alpha [\text{Subj} [H YP]]]]$
   b. i. there are likely to be awarded several prizes
       ii. several prizes are likely to be awarded
   c. i. we expect there to be awarded several prizes
       ii. we expect several prizes to be awarded

The Case/agreement properties of Subj in (4a), and its overt location, are determined by properties of the matrix probe $P$, not internally to $\alpha$. $\alpha$ is a TP with defective head $T_{def}$, which is unable to determine Case/agreement but has an EPP-feature, overtly manifested in (4c). Raising-ECM parallels give good reason to believe that the EPP-feature is manifested in (4b) as well, by trace of the matrix subject; preference for Merge over (more complex) Move gives a plausible reason for the surface distinction between [Spec, $T_{def}$] in (4b) and in (4c) (see MI). In (4bi) and (4ci), the EPP-feature of $T_{def}$ is satisfied by Merge of expletive; in (4bii) and (4cii), by raising of the direct object.

excerpt from Derivation by Phase

Grammar formalisms e.g. Stabler (1998), (2011) (Lecomte & Retoré) 2001
What constitutes an implementation?

*actually, should need agree(t,a) as well to value Case on PRT

one probe, multiple goals: matrix T agrees with 3 goals

implementation optimizes this away: 

\textit{uCase can be unified during agree}

*agree(t,n) 
  \( t: u\phi \), Nominative 
  \( n: \phi, \text{uCase} \)

*agree(t,n) 
  \( t: u\phi \), Nominative 
  \( n: u\phi \)

*agree(t\_def,n) 
  \( t\_def: u\phi \) 
  \( n: \phi, \text{uCase} \)

*agree(a,n) 
  \( -ed: u\phi, \text{uCase} \) 
  \( n: \phi, \text{uCase} \)

one probe, multiple goals: matrix T agrees with 3 goals: matrix T agrees with 3 goals

*agree(t,n) 
  \( t: u\phi \), Nominative 
  \( n: \phi, \text{uCase} \)

*agree(t,n) 
  \( t: u\phi \), Nominative 
  \( n: u\phi \)

*agree(t\_def,n) 
  \( t\_def: u\phi \) 
  \( n: \phi, \text{uCase} \)

*agree(a,n) 
  \( -ed: u\phi, \text{uCase} \) 
  \( n: \phi, \text{uCase} \)
What constitutes an implementation?

Same number of Merge steps as before

Probe [a!case ed] agrees with goal [n!case several prizes]
Probe [tdef] agrees with goal [n!case several prizes]
Probe [t] agrees with goal [n several prizes]
Computational complexity?

$\alpha \#\text{merge steps} + \beta \#\text{nodes probed} + ...$
Minimal Computation

- overriding condition of **Minimal Computation** (MC) means design for the **Generative Procedure** (GP):
  - “whenever efficiency of design and communication conflicts: design wins”
- no **Parsing Procedure** (PP) is specified
  - parsing need not be efficient
- **Example:**
  - several prizes are likely several prizes to be awarded several prizes

suggest consequences for cognitive architecture. Keeping to the principle of MC, the GP yields forms that are appropriate for semantic interpretation at CI but not for production and perception at sensory-motor (SM), though this SM inadequacy also follows from MC. The inadequacy is severe. Anyone who has worked on parsing programs knows that a major difficulty is posed by “filler-gap” problems: given *who* in (10), the problem for parsing/perception is to locate the gap where it receives its interpretation in the argument structure of the sentence, not a trivial matter in general. These problems would largely be obviated if all copies were pronounced, violating MC.

Chomsky (2011)
Design Minimalism

Generative Procedure (GP)

• Design minimalism:
  – good for complexity, but what about data?

• Limits on derivation
  conservation of syntactic objects (SO)
  – e.g. only copies, can’t create indices

• Problem for modules
  e.g. Binding theory (BT)
  – no Free Indexation
  – no theory-internal levels of representations
  – no Binding Conditions A, B and C

assuming we still want to have a syntactic BT...
Growing the implementation

**Binding theory** *(joint work with Jason Ginsburg, U. of Aizu, Japan)*

- Idea:
  - doubling constituent (DC), adapted from Kayne (2002)
  - e.g. [he John]

\[
\begin{array}{c}
\text{TP John thinks TP [he John] [v is AP smart [he John]]]}
\end{array}
\]

John thinks John he is smart John he

- Phases: derive distributional differences between pronouns and anaphors

Related work:
- Zwart (2002)
John$_i$ thinks he$_i$ is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>theta, Case</td>
<td>theta, Case</td>
</tr>
<tr>
<td>licensed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
John \_i\_ thinks he \_i\_ is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>Case</td>
<td>theta, Case</td>
</tr>
<tr>
<td>licensed</td>
<td>theta</td>
<td></td>
</tr>
</tbody>
</table>
John \_i \text{ thinks he} \_i \text{ is smart}

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>Case</td>
<td>theta, Case</td>
</tr>
<tr>
<td>licensed</td>
<td>theta</td>
<td></td>
</tr>
</tbody>
</table>
John\textsubscript{i} thinks he\textsubscript{i} is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>theta, Case</td>
<td></td>
</tr>
<tr>
<td>licensed</td>
<td>theta, Case</td>
<td></td>
</tr>
</tbody>
</table>
John_{i} thinks he_{i} is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>theta, Case</td>
<td></td>
</tr>
<tr>
<td>licensed</td>
<td>theta, Case</td>
<td></td>
</tr>
</tbody>
</table>
John\textsubscript{i} thinks he\textsubscript{i} is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>theta, Case</td>
<td></td>
</tr>
<tr>
<td>licensed</td>
<td>theta, Case</td>
<td></td>
</tr>
</tbody>
</table>
John\textsubscript{i} thinks he\textsubscript{i} is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>theta, Case</td>
<td></td>
</tr>
<tr>
<td>licensed</td>
<td>theta, Case</td>
<td></td>
</tr>
</tbody>
</table>

**Essential idea:**
probe-goal search has limited range
Suppose r-expr with unvalued uninterpretable feature at the limit of search is allowed to undergo theta Merge ...

*obviously, timing is critical*
John$_i$ thinks he$_i$ is smart

<table>
<thead>
<tr>
<th>scoreboard</th>
<th>he</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>lacks</td>
<td>Case</td>
<td></td>
</tr>
<tr>
<td>licensed</td>
<td>theta, Case</td>
<td>theta</td>
</tr>
</tbody>
</table>

Last Resort (LR): theta Merge

Note: needed for all theories with a Doubling Constituent (DC)
John_i thinks he_i is smart

Probes [t] agrees with goal [d[d][n john]]
John\textsubscript{i} thinks he\textsubscript{i} is smart

Spell-Out
John John thinks he John is smart he John thinks he is smart
Data

• Analysis of classic BT data:
  – *John_i praises him_i
  – John_i praises himself_i
  – John_i thinks he_i is smart
  – *He_i thinks John_i is smart
  – *John_i thinks himself_i is smart
  – *John_i thinks that Mary likes himself_i
  – John_i considers himself_i to be intelligent
  – *John_i considers him_i to be intelligent
  – John_i likes his_i dog
  – *John_i likes himself_i’s dog
  – ?*Hannah_i found a picture of her_i
    – Hannah_i found a picture of herself_i
  – ?*Hannah found Peter_i’s picture of him_i
    – Hannah found Peter_i’s picture of himself_i
    – Hannah found Peter_i’s picture of her_i
    – Hannah found Peter_i’s picture of herself_i
    – etc.

assume the DC for anaphors is different to begin with
[\_D \_D self][\_N he John]]

phase boundary

step-by-step examples @ http://dingo.sbs.arizona.edu/~sandiway/mpp/binding/examples/
also paper
More Data

- **double objects**
  - I showed John to himself in the mirror
  - *I showed himself to John in the mirror
  - I showed John himself in the mirror
  - *I showed himself John in the mirror

step-by-step examples @
http://dingo.sbs.arizona.edu/
~sandiway/mpp/binding2/examples/
also poster (TCP 2012)
http://dingo.sbs.arizona.edu/
~sandiway/mpp/TCP2012.pdf
Even More Data

Spell-Out
John is tough [Op John]-PRO to PRO please
[Op John]
John is tough to please

DC for tough movement
Hicks (2009)
[Op John]
(Op = empty operator)
“smuggling operation”

step-by-step example @
http://dingo.sbs.arizona.edu/~sandiway/mpp/hicks/examples/
tentative analysis only ...
Computational Complexity

Phases

• Long distance:
  – $\text{John}_i$ thinks that $\text{Peter}^{*i}$ thinks that $\text{Mary}$ thinks that $\text{Bill}^{*i}$ likes $\text{him}_i$

• Cost of locality:
  – probe-goal search is local
  – iterated movement to the edge of a Phase

coreference possibilities:
[he John] preferred over [he Peter]
Phases

without a Buffer, would need iterated movement to the edge of each marked phase

matrix T will attract John

Computational complexity
α #merge steps + β
#nodes probed + ...

phase boundary
Computational Complexity

Phases

• Long distance:
  – John\(^i\) thinks that Peter\(^*\(_i\) thinks that Mary thinks that Bill\(^*\(_i\) likes him\(_i\)

• Cost of locality:
  – probe-goal search is local
  – iterated movement to the edge of a Phase

• Alternate implementation
  – eliminate iterated movement of this sort (smaller trees)
  – use a Buffer (for theta Merge)
  – no extra ambiguity
    – Preference: external Merge < internal Merge < Buffer
Generative and Parsing Procedures

• Idea:
  – Generative Procedure (GP) and Parsing Procedure (PP) are not separate devices
  – but GP has priority over PP (Minimal Computation)
Generative and Parsing Procedures

1. theta merge V & N
2. merge PRT & V
3. merge v & A
4. merge T & v
5. move to spec-T
6. merge A & Tdef
7. merge v & A
8. merge T & v
9. move to spec-T
10. merge C & T

sequence of Merges can be had from an in-order traversal but reporting Merges on the way back up...
Generative and Parsing Procedures

Two problems peculiar to parsing that must be dealt with:

1. displacement without all copies being pronounced
2. empty heads (t, v and others)

In-order traversal:
- top down
- left to right expansion

Probe [a!case ed] agrees with goal [n!case several prizes]
Probe [tdef] agrees with goal [n!case several prizes]
Probe [t] agrees with goal [n several prizes]
Parsing Procedure (PP)

• **Implementation summary**
  1. left-to-right, incremental expansion of the covering grammar:
     • no need to recode GP
  2. probe-goal: GP, PP *nearly the same*
     • **timing issues**: active/inactive goals
  3. displacement: must put copies in gap positions
     • need some memory device: Buffer
     • potential ambiguity (fill from input or Buffer)
  4. empty heads: e.g $T/T_{\text{def}}, v/v^*$
     • PP (ambiguity), GP (no ambiguity, part of numeration)
Several prizes are likely to be awarded

At this point we know there is a copy of *several prizes* downstream *but where exactly.. we don’t yet have enough information*

**Note:**
- introduced probe T
- *its c-command domain doesn’t exist yet...* freeze probe-goal search
Several prizes are likely to be awarded
Several prizes are likely to be awarded

Option 1: copy Buffer into open lower spec-T matrix T probe finds its goal

Note: introduced probe $T_{\text{def}}$ freeze probe-goal search
Several prizes are likely to be awarded

alternate choice:
don’t drop several prizes into subject position
Several prizes are likely to be awarded
Several prizes are likely to be awarded

Note:
introduced probe PRT –ed
freeze probe-goal search

actually..
two out of the three probes are concurrently operating
Several prizes are likely to be awarded.
Several prizes are likely to be awarded

alternate choice: don’t drop several prizes into open object position

probes $T_{\text{def}}$ and PRT still pending
Several prizes are likely to be awarded.

step-by-step examples @ http://dingo.sbs.arizona.edu/~sandiway/mpp/cuny2012/examples/
Summary

- **theory**: Chomsky (2001+ ...)
- **implement**: merge, agreement, probe-goal search, phases
- GP and PP share architecture:
  - traverse grammar in same way
  - same probe-goal algorithm
  - MC: we need a memory device to handle displacement
  - PP buffer shared

*GP: buffer used in DC analysis for pronoun binding*

*improves GP from the standpoint of MC*
Appendix
Principles and Parameters Approach

PAPPI: Fong (1990)
Principles and Parameters Approach

Generate and test:
33 candidate parses examined by the parser

most are eliminated early by Case/Theta theory sub-system