A Relabeling Analysis of English Possessives*

Jason Ginsburg
Osaka Kyoiku University
jginsbur@cc.osaka-kyoiku.ac.jp

Sandiway Fong**
University of Arizona
sandiway@email.arizona.edu

*Supported by Japan Society for the Promotion of Science Grant-in-Aid for Scientific Research #16K02769

**Currently on leave at Kyoto University
Outline

• Data
• Core Assumptions
• Combinatorics (the model)
• Examples
• Conclusion
Data
English possessive DPs

• In English, *of*-insertion is regular but there is variation in the deployment of the double genitive.

(1)

a. my friend/the friend of mine/*the friend of mine’s
b. your friend/*the friend of your/the friend of yours
c. his friend/the friend of his/*the friend of his’(s)
d. her friend/*the friend of her/the friend of hers
e. their friend/*the friend of their/the friend of theirs
English possessive DPs

How do possessives work?
(2) my friend
• Assume that this is a DP with head ‘s. D needs Case (has uCase), but D also checks uCase on friend
(3) the friend of mine
• friend is the underlying object of my
• Why isn’t friend pronounced in object position?
• There is a possession-type relation between my and friend.
• Compare (4a-b) (cf. Barker 1998). In 4a, John owns the picture. In (4b), the picture is of John.
(4) a. a picture of John’s hangs in the gallery
  b. a picture of John hangs in the gallery
• Why don’t you say any of these:
(5) a. *the friend of my’s
  b. *the friend of mine’s
English possessive DPs

• There is variation in the deployment of double genitives

(6) the friend of mine vs. *the friend of mine’s
(7) *the friend of your vs. the friend of yours
(8) the friend of his vs. *the friend of his’s
(9) *the friend of her vs. the friend of hers
(10) *the friend of their vs. the friend of theirs

• Assume that PF rules are at work.
  • For example, in (6) mine blocks mine’s, in (7) yours blocks your
  • my + ‘s + friend = mine (my + ‘s + friend ≠ mine’s, my’s)
Core Assumptions
Proposal

• Theoretical underpinnings:
  • Minimalist framework of Chomsky (2013) – free Merge, labeling
  • Cecchetto and Donati’s (2015) relabeling proposal (for relative clauses)

• Result:
  • We show how target examples (English possessives) can be **computed**
Merge

• Merge is free (Chomsky 2004, 2005, 2013, 2015)
  • no feature-driven movement

• Internal Merge (IM) and External Merge (EM) are free
  • IM and EM are both freely available*

External Merge of X with Y

Internal Merge of Y with X

* A Chomsky 2017 lecture (University of Arizona) suggests IM is preferred over EM for minimal search reasons. Also see Shima (2000).
Set Merge: Labeling

(a) Head X labels
(b) Head X is too weak to label unless strengthened
(c) No label
(d) Y labels if XP moves out
   • Not all copies of XP are within this Syntactic Object (SO)
   • Y is not weak

• External Set Merge is free
• Internal Set Merge is free
Strengthening

- R is weak
- In (a), categorizer X labels
- In (b), phase head y* transmits uPhi (and Case valuing) to R.
  - Agree(R,XP) checks uPhi on R
  - <ϕ,ϕ> labels, as R and XP have identical ϕ-features
  - strengthened R may label {R,XP} (* represents strengthening)
- In (c), n* strengthens R

(a) 
```
  X
 /   \
X     R
```

(b) 
```
  y*
   /\  
  <ϕ,ϕ>
```

(c) 
```
  n*
   /\  
  R   XP
```
Pair Merge

PM is asymmetric Merge
Y is on a separate plane (Y not visible to SELECT, nor AGREE, nor labeling).

(a) Y is externally PM’ed with ZP

(b) Y is internally PM’ed with XP

Internal and external PM are free (cf. Richards 2009, Epstein, Kitahara, & Seely 2016)
Combinatorics (the model)
Towards Efficient Computation

• Recursive Free Merge can be a computational nightmare, but how bad is it?
  • free Merge means displacement is not feature-driven (old EPP/Edge feature)

• Faculty of Language (3 factors), Chomsky (2005):
  • Genetic endowment: Merge, Labeling etc.
  • Experience
  • Third Factor (incl. principle of efficient computation)

• Possible limits on Merge
  • filtered by Labeling (CI interface transfer: triggered at a phase)
  • selection (root + categorizer)
  • efficient computation: e.g. no duplicate structures, no infinite loops, etc.
Unrestricted Set Merge

• Consider two heads r and n: How many way to merge them?

  recursively apply:

  **ESM**: External Set Merge
  Merge current SO with some other SO in the workspace

  **ISM**: Internal Set Merge
  SELECT a (proper) sub-SO of the current SO to merge to itself

  **Initial workspace (WS)**: \( r, n \)

  • 1 Merge:
    • \{r, n\}

  • 2 Merges
    • \{'r,\{r, n\}\}, \{n,\{r, n\}\}\

  • 3 Merges
    • \{{r, n},\{r, \{r, n\}\}\}, \{n,\{r, \{r, n\}\}\},
      \{r,\{r, \{r, n\}\}\}, \{{r, n},\{n, \{r, n\}\}\},
      \{n,\{n, \{r, n\}\}\}, \{r,\{n, \{r, n\}\}\}\}

  • *etc..*
Unrestricted Set Merge

• Consider two heads r and n:
  How many way to merge them?
  • y-axis ☞: $\log_{10}(\#\text{SOs})$

Answer:
without duplicates,
$\#\text{SOs} = n!$
$(n=\#\text{Merges})$
Free Merge Constraints

• In just three steps, Internal Set Merge (ISM) can create **duplicates SOs**

• **Example**: in \{r,\{r,n\}\}
  by selecting either copy of r,
  ISM can create same SO \{r,\{r,\{r,n\}\}\}

A possible 3\textsuperscript{rd} factor constraint: **no duplicate SOs**

• can be eliminated cheaply
  • **Memorization**: i.e. *must be able to spot duplicates locally*

• Derivation Tree:
  Start: SO: \(r\), Input: \([n]\)
  1. ESM, SO: \{r,n\}, Input: \([]\)
  1.1. ISM, SO: \{r,\{r,n\}\}, Input: \([]\)
    1.1.1. ISM, **SO**: \{r,\{r,\{r,n\}\}\}, Input: \([]\)
    1.1.2. ISM, **SO**: \{\{r,n\},\{r,\{r,n\}\}\}, Input: \([]\)
    1.1.3. ISM, **SO**: \{r,\{r,\{r,n\}\}\}, Input: \([]\)
    1.1.4. ISM, **SO**: \{\{r,n\},\{r,\{r,n\}\}\}, Input: \([]\)
  1. 2. 1. ISM, **SO**: \{n,\{r,\{r,n\}\}\}, Input: \([]\)
    1.2.2. ISM, SO: \{\{r,n\},\{n,\{r,n\}\}\}, Input: \([]\)
    1.2.3. ISM, **SO**: \{r,\{n,\{r,n\}\}\}, Input: \([]\)
    1.2.4. ISM, **SO**: \{n,\{n,\{r,n\}\}\}, Input: \([]\)
Free Merge Constraints

• Eliminating duplicate SOs:
  • X-axis: number of Set Merges (SM)
  • Y-axis \(\uparrow\): number of SOs built
  • Y-axis \(\downarrow\): log number of SOs built
  • Orange line: allowing duplicates
  • Blue line: with no duplicates

No duplicates means a slightly more powerful ISM proof system
Restrictions on Internal Merge

• So far just one kind of restriction: no duplicate SOs
  • e.g. as shown earlier: \{c,\{a,\{b,a}\}\} = 2 ways => \{a,\{c,\{a,\{b,a\}\}\}\}
  • of convenience to the proof system (only?)

• Another kind: block repetitive operations (loops)
  • enlarge SO without bound (no decrease in WS size)
  • *πππ where π is an IM sequence
    • e.g. ISM(a) ISM(a) or ISM(a) ISM(b) ISM(a) ISM(b) etc.
  • Suppose FL always (attempts to) block infinite loops (computational minimalism)

• Yet another kind: lemmas: (can be applied proactively)
  • uF = unvalued feature F
  • e.g. can’t PM β[uF] to α forming <β, α>, where β is an adjunct
  • since β is no longer accessible to operations, β[uF] can never get valued

Only one kind of repetition permitted (i.e. none), e.g. no rule *π₅⁺ (i.e. you can repeat up to 4 times but not more)
Combinatorics: \{\text{the}, \text{d}\}, \{\text{book}, \text{n}\}

- Only convergent thread in the finite computation tree!
- Statistics:
  - Nodes: 41
  - Loops detected: 13
  - Duplicates: 0
  - PMergeR: 15
  - MergeR: 53
  - Unlabeled: 17
  - Derivations: 1
  - Max # of merges: 8. Derivations completed.
Combinatorics: *the friend of mine*

14 Operations
### Combinatorics: the friend of mine

**Heads:** [friend,n!case,[me,n!case,'s,d*],n*!case,[the,d]]

<table>
<thead>
<tr>
<th>Merge #</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Branching factor**

- Orange = Successful derivation
- (SO shown on previous slide)
Examples
Derivations

- SM and PM are free
- EM and IM are free
- Labeling occurs at the phase level
  - Complement of phase head is transferred
- Phase head transfers inflectional features to next lower head
- Rules of phonological form apply at transfer (after derivation is complete)
  - We assume no countercyclic operations (e.g. head-movement)
    - (some) head-movement phenomena can be relegated to Phonological Form (PF)
Derivations

• Merge Restrictions:
  (a) roots must be categorized (as soon as possible)
  (b) each categorizer must find its root (with no intervening heads)
  (c) categorizers can only categorize once
    e.g. *{c,{R,{c,R}}}) formed with only c and R (R=root, c=categorizer)
  (d) can’t PM $\beta[uF]$ to $\alpha$ forming $<\beta,\alpha>$, where $\beta$ is an adjunct
    since $\beta$ is no longer accessible to operations, $\beta[uF]$ can never get valued

• CI Interface Restriction:
  (a) *<nP,nP>, *<dP,dP>
    • Interface expects <dP,nP>
my friend

- SM me (root) & n
- SM ‘s (root)
- SM (internal Set Merge) {me, n}
my friend

- SM *me* (root) & n
- SM ‘s (root)
- SM (internal Set Merge) {me, n}
- SM d*
  - uPhi and inherent Case are passed down to root ‘s
- Agree(‘s,n)
  - uPhi checked on ‘s
  - uCase checked on n
- d* (phase head) triggers transfer
  - Labeling occurs
  - Shared φ label
    - Shared φ strengthen ‘s \(\Rightarrow\) ‘s*
- SM friend & n
  - !case = uninterpretable Case
  - n will label at transfer
- PM my & friend
- Spell-Out
  - me ‘s friend \(\Rightarrow\) my friend
the friend of mine

- form *my friend*
- PM *my friend & friend*
  - internal PM of *friend*
  - *my friend* = adjunct
- SM n*
  - ability to assign inherent Case is passed down to *friend*
- Agree(*friend*, n) → n gets inherent Case, pronounced as *of*
- SM d & *the*
- PM {*the*, d} & friend of *my*
  - {*the*, d} = adjunct
the friend of mine

• Spell-Out is tricky
  • Note that the elements of the tree aren’t ordered yet

• n has inherent case – spelled out as of preceding my friend

• me ‘s friend n = mine

the d friend InherentCase me ‘s n friend = the friend of mine
Conclusions

• An account of target possessive constructions in a featureless free-Merge system
• A computer model computes all possible structures (*perhaps a first*)
• However, overgeneration can cause issues:
• Examples:
  (11) *the friend the
  (12) *the friend of the
    • Perhaps assume that *the* (and *a*) can never be stranded in English
• Other puzzle:
  (13) #Mary’s friend of yours
    • on the intended reading: *Mary’s friend & friend of yours*
    • a problem to be resolved past CI Interface’s door?
Spotted yesterday...
References


