LING 696G: Lecture 1

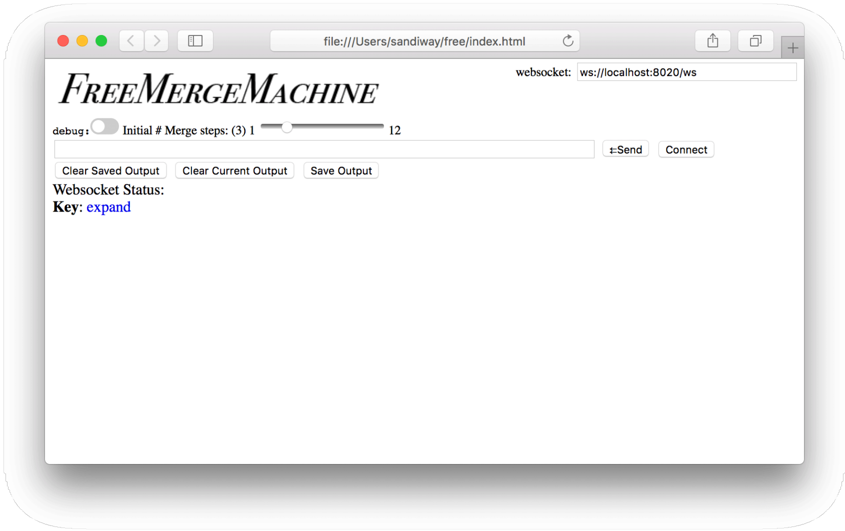
Sandiway Fong
(joint work with Jason Ginsburg, Osaka Kyoiku University)

Topics

This is an advanced class:

this year we'll do new stuff …

- Computation:
  - Serial and parallel computation
- Linguistic Theory:
  - Set/Pair Merge (Chomsky, recent work)
  - Merge is not feature-driven but "free"
  - Labeling
- Implementation:
  - Free Merge Machine
  - Underlying technology:
    - Javascript
    - Websockets
    - Prolog (me); Python (Jason)



Deliverables

- A term project either based on the Underlying Technology (previous slide) or on the Free Merge Machine (FMM)

Readings

Read and be ready to discuss next week:

- *The Hunt for a Label*, Oishi (2015)
    - NPs not DPs ..

Background:

- *Approaching UG from Below*, Chomsky (2007)
- + work thereafter

Egashira, Hiroki et al. eds _In Untiring Pursuit of Better Alternatives._ 222-334. Kaitakusha, Tokyo, March, 2015.

The Hunt for a Label*

Masayuki Oishi
Tohoku Gakuin University

1. Lexicalist Position

One of the alleged merits of the X-bar theory was that it is successful in capturing cross-categorial generalizations with the crucial stipulation of endocentricity, such as internal structure between nominal and sentential expressions. The X-bar theory was claimed to serve to assign one and the same underlying skeletal structure to the expression containing, say a particular verb on the one hand and to that with its derived nominal on the other, with a crucial assumption that these two categories are listed under a single lexical entry. Take a case of a verb *decide* and its derived nominal *decision*. The distinction between *decide* and *decision* is just a matter of phonological realization of a set of features, *DECIDE*, with the selectional properties common to them.

This sense coincides with the general reflection on syntactic aspects of lexical items, or substantive elements. Specifically, it is the idea that syntactically derivable properties are not stated in the lexicon and hence a category label, which is quite syntactic, is automatically determined in syntax. Here "automatic" determination awaits certain clarifications. Thus, earlier work of linguistic theory had claimed that once a category-neutral element DECIDE is inserted under N in the base, it is treated as an N (*decision*), and no transformation (or syntactic operation) can change its category

* The paragraphs to follow are meant to constitute a preliminary sketch of my on-going research on labeling in nominal structures (Oishi (in preparation)). I owe Noam Chomsky for detailed discussion and encouragements and Hisatsugu Kitahara for clarifications and comments, whom I thank deeply. Some materials here have been discussed in Oishi (2005, 2011, 2012).

Merge

- Lexical Items (LIs): a, b
- Set Merge {A,B}
  - symmetric
- Pair Merge <A,B>
  - ordered pair: asymmetric
  - e.g. A is adjunct to B
- Recursive:
  - output of Merge can be input to Merge
- Free Merge
  - Merge is not feature-driven
  - e.g. EPP, Edge features
- Workspace (WS)
  - multiple WSs possible

Recursive Free Merge

- Recursive Free Merge can be a nightmare computationally, 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

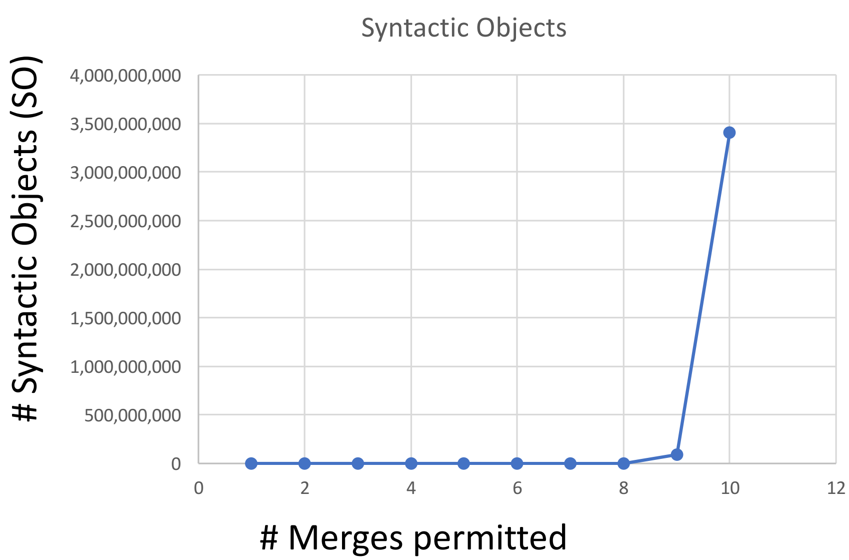
- Consider two heads *r* and *n*: How many way are there to merge them?

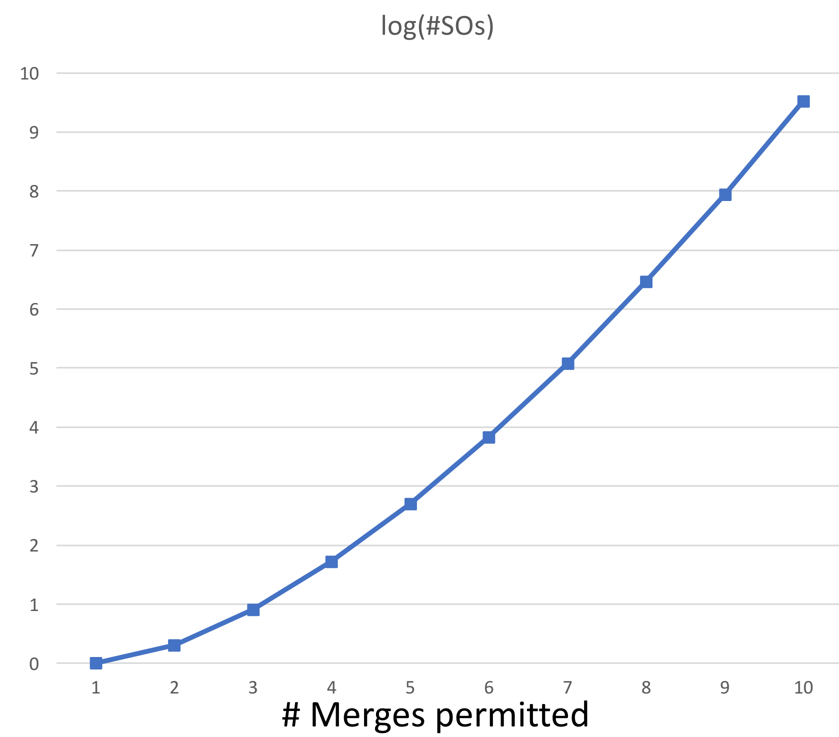
- Initial workspace: r, n
- pick one of them to be the current Syntactic Object (SO)
- and recursively apply:
  - **ESM**: External Set Merge
    - Merge current SO with some other SO in the workspace
  - **ISM**: Internal Set Merge
    - **SELECT** a (proper) sub-SO of the current SO to merge to itself

- 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 ☞: $\log_{10}$(#SOs)





Set Merge {r,n}

- In just three steps, Internal Set Merge (ISM) with blind selection can create **duplicates SOs**

- **Example**: in {r,{r,n}}

  by selecting either copy of r,

  ISM can create same SO {r,{r,{r,n}}}

- 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: {n,{r,{r,n}}}, Input: []

1. 2.ISM, SO: {n,{r,n}}, Input: []

1. 2. 1.ISM, SO: {n,{n,{r,n}}}, Input: []

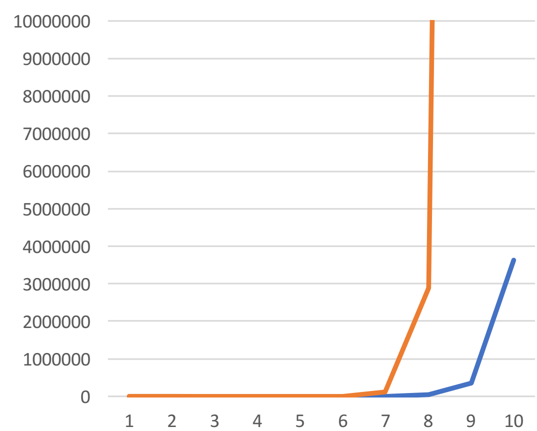
1.2.2.ISM, SO: {{r,n},{n,{r,n}}}, Input: []

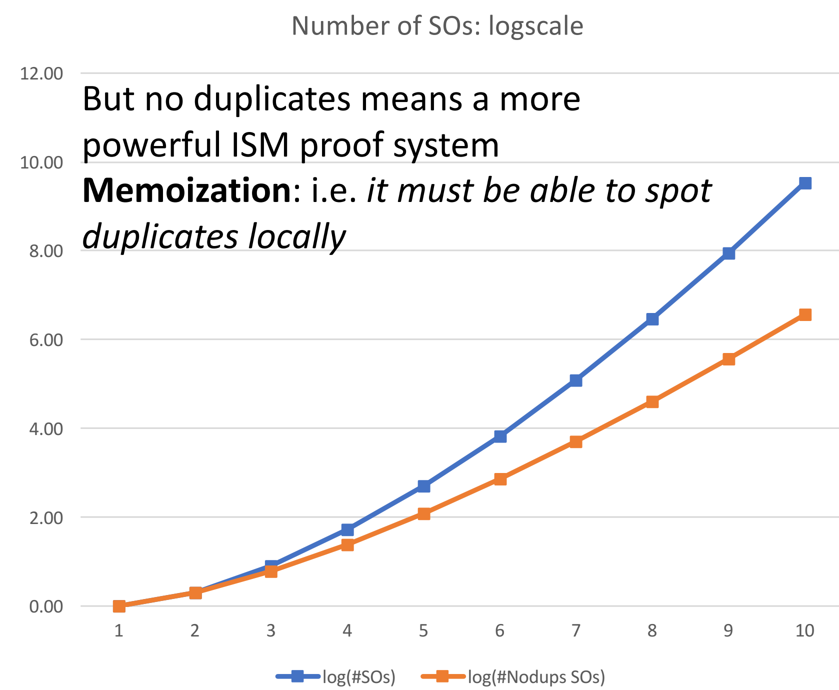
1.2.3.ISM, SO: {r,{n,{r,n}}}, Input: []

1.2.4.ISM, SO: {n,{n,{r,n}}}, Input: []

Duplicates vs. No Duplicates

- Eliminating duplicate SOs:
  - X-axis: number of Set Merges (SM)
  - Y-axis ☝ : number of SOs built
  - Y-axis ☞: **log** number of SOs built
  - Orange line: allowing duplicates
  - Blue line: with no duplicates





Unrestricted Set Merge

Back to original question…

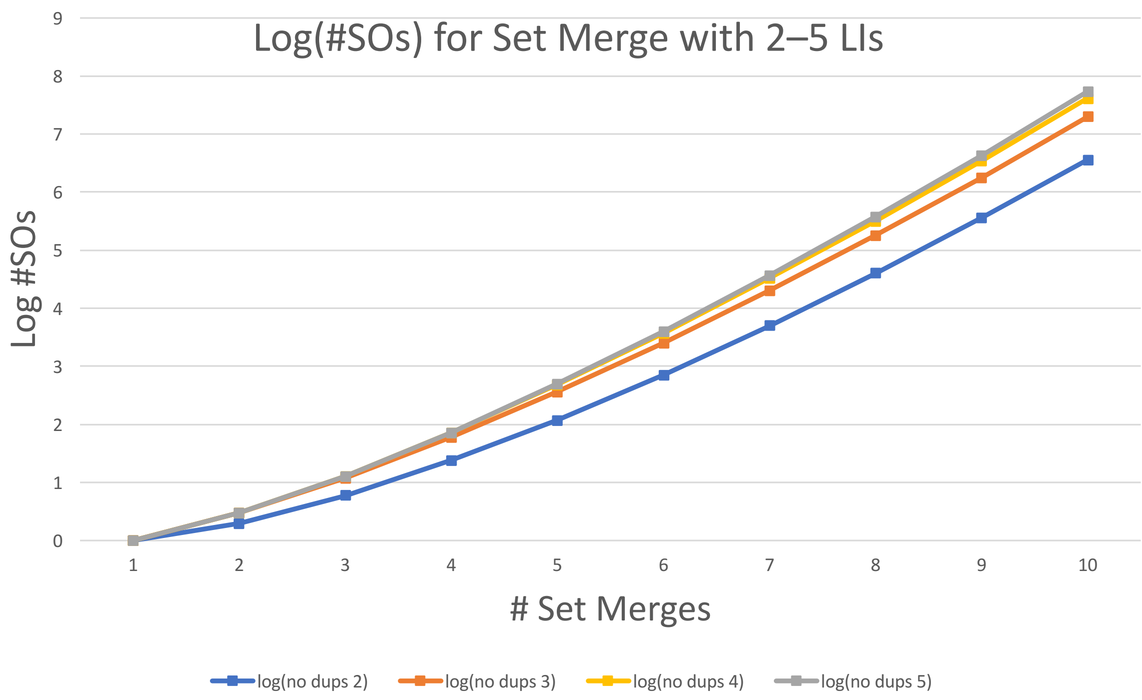
- Consider two heads r and n: How many way to merge them?
- **Answer**: without duplicates, it grows as *n*! (*n*=# Merges)
- Proof:
  - **Sketch**: suppose SELECT has choice of *k* sub-SOs for ISM. Then next round, it will have *k+1* choices.
  - **Why**? select will have all the same choices + 1 (the complete SO last round will now be a sub-SO)

# Set Merges	no duplicates
1	1
2	2
3	6
4	24
5	120
6	720
7	5,040
8	40,320
9	362,880
10	3,628,800

Unrestricted Set Merge: computed

- Number of heads:
  - 2: [a,b]
  - 3: [a,b,c]
  - 4: [a,b,c,d]
  - 5: [a,b,c,d,e]

- (*This model assumes that each time only the current SO is being added to; this is a restricted case*.)



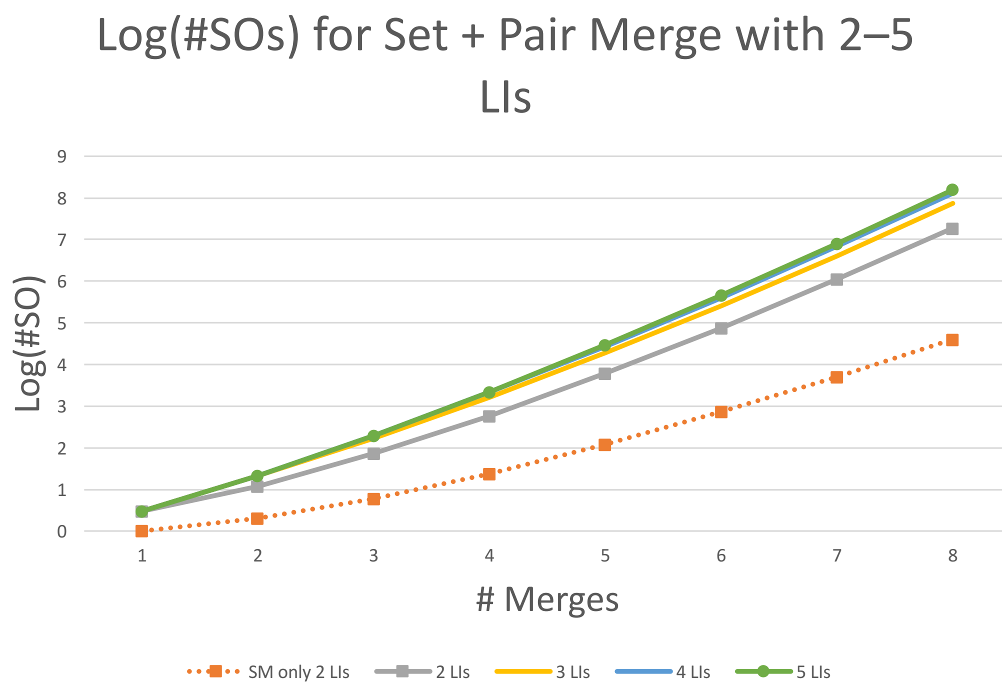
Unrestricted Set Merge: computed

- More heads: 3,4,5…

# SM	no dups 2	no dups 3	no dups 4	no dups 5
1	1	1	1	1
2	2	3	3	3
3	6	12	13	13
4	24	60	72	73
5	120	360	480	500
6	720	2520	3720	4020
7	5040	20160	32760	36960
8	40320	181440	322560	381360
9	362880	1814400	3507840	4354560
10	3628800	19958400	41731200	54432000

Set + Pair Merge: computed

- Types:
  - External Set Merge (ESM)
  - Internal Set Merge (ISM)
  - External Pair Merge (EPM)
  - Internal Pair Merge* (IPM)



*without certain restrictions