

LING/C SC 581:

Advanced Computational Linguistics

Lecture 29

Prof. Sandiway Fong

Continuing on from last time

- So far:
 - **Biology:**
 - sensors are amazing, but we don't need the resolution
 - **Neurons:**
 - artificial ones vs. nature's chemistry-limited cells
 - **Third Factor:**
 - computational efficiency and bandwidth important for brain organs
 - **Merge:**
 - binary Merge is the simplest possible structure-building operation

Aside: words *don't* refer to things (*contra* Quine)

As they step into the same rivers, ever-different waters flow (B 12; Heraclitus)

Words
refer to
mental
Concepts

[planes]



planes



you can call it pragmatics if you want

Merge as a primitive operation

- Recall there are simpler things than Simplest Merge
 - e.g. linear concatenation (Chomsky 1956 Model #1)
 - e.g. **non-recursive** (so-called) Core-Merge, Fujita (2014) ← *same as concat-2?*
 - e.g. Conjoin, Progovac (2015)
 - e.g. recursive Internal Merge (IM) only (*as it's more efficient*)
 - **NO**: *can't work on its own, it can't get started*
 - e.g. recursive External Merge (EM) only
 - hmm, can build θ -configurations now, but all languages exhibit displacement
- *Posit them as earlier stages of evolution?*
 - *only necessary if (you believe) recursive Merge can't emerge in "one step"*

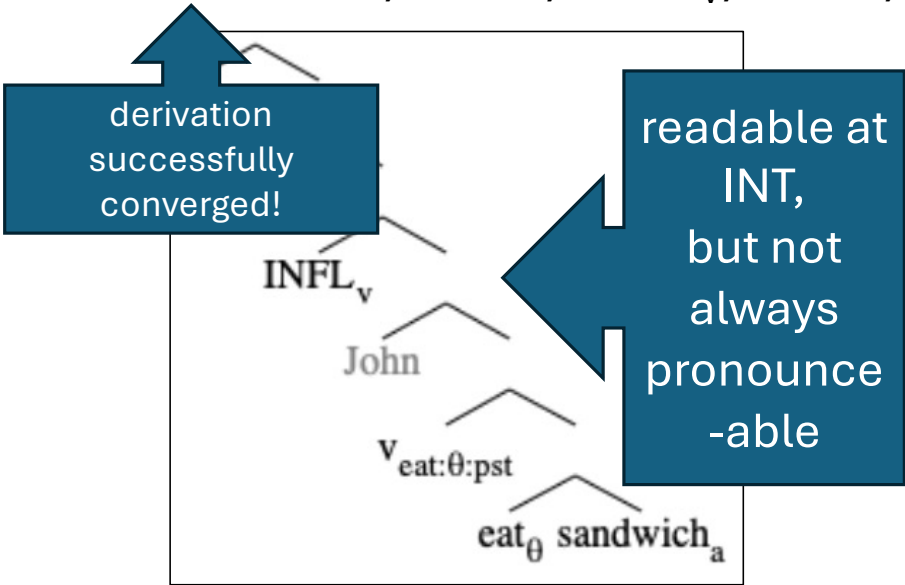
SMT Parser

Words: John ate a sandwich

Initial WS 1: sandwich_a eat_θ v_{eat:θ:pst} INFL_v John

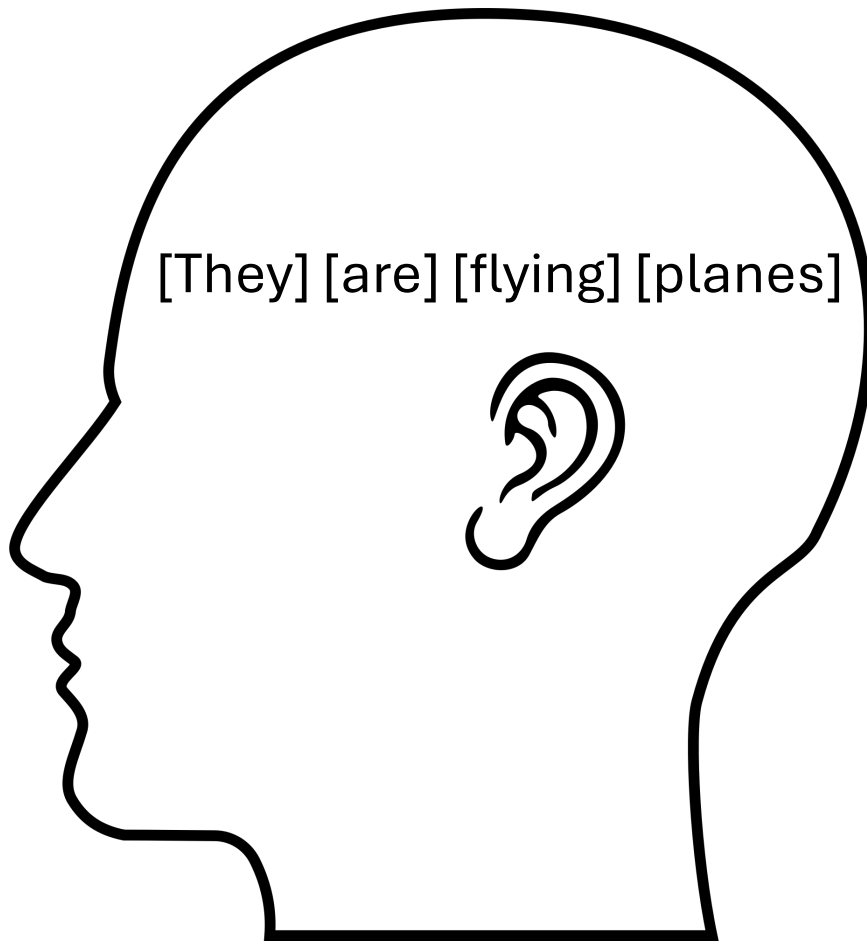
ONE OBJECT: {C, {John, {INFL_v, {John, {v_{eat:θ:pst}, {eat_θ, sandwich_a}}}}}}

a sandwich = sandwich_a



Words: sandwich a ate John

- ▼ Initial WS 1: John_a eat_θ v_{eat:θ:pst} INFL_v sandwich
- ▶ WS 1: {eat_θ, John_a} v_{eat:θ:pst} INFL_v sandwich
- INT/EXT: \wedge {C, {sandwich, {INFL_v, {sandwich, {v_{eat:θ:pst}, {eat_θ, John_a}}}}}}
- ▶ Initial Spellout: sandwich 3sg pst eat a John
- Error:** inconsistent spellout
- ▶ WS 1: {eat_θ, John_a} v_{eat:θ:pst} INFL_v sandwich
- ▶ WS 1: {eat_θ, sandwich} John_a v_{eat:θ:pst} INFL_v
- INT/EXT: \wedge {C, {John_a, {INFL_v, {John_a, {v_{eat:θ:pst}, {eat_θ, sandwich}}}}}}
- ▶ Initial Spellout: a John 3sg pst eat sandwich
- Error:** inconsistent spellout
- ▶ WS 1: {eat_θ, sandwich} John_a v_{eat:θ:pst} INFL_v
- ▶ WS 1: {v_{eat:θ:pst}, eat_θ} John_a INFL_v sandwich
- ▶ WS 1: {eat_θ, v_{eat:θ:pst}} John_a INFL_v sandwich

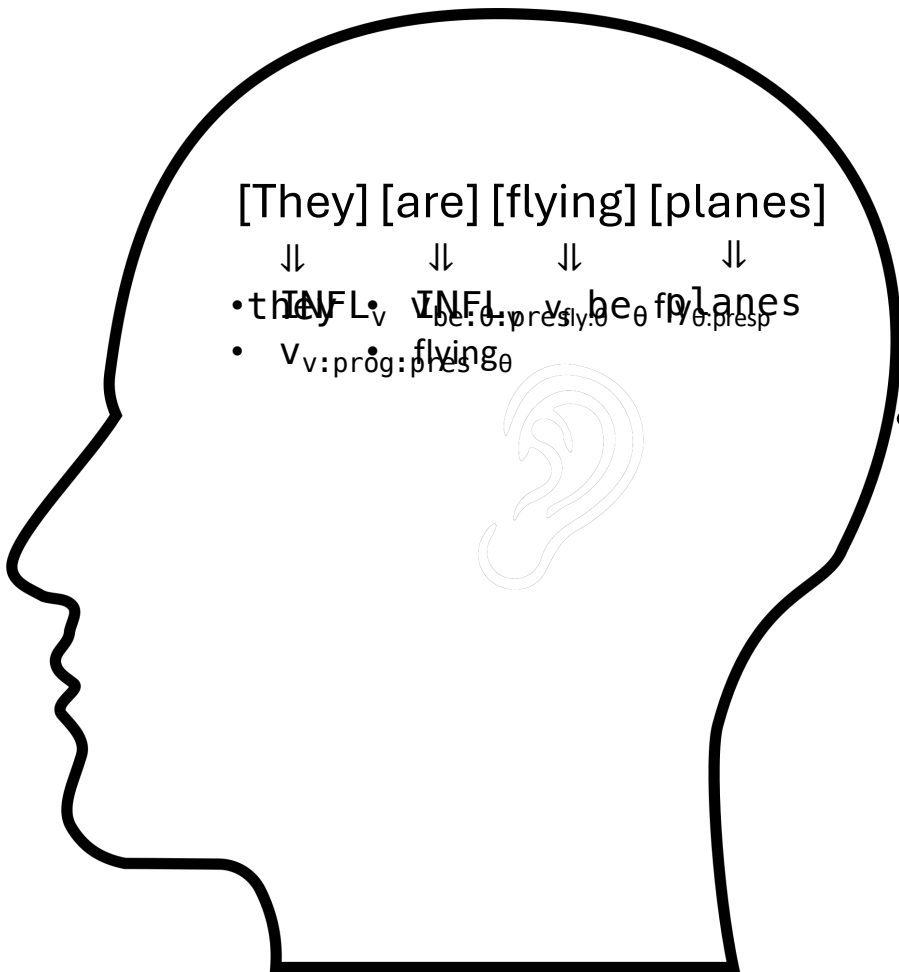


How it works

[pg.118, Chomsky (1956)]

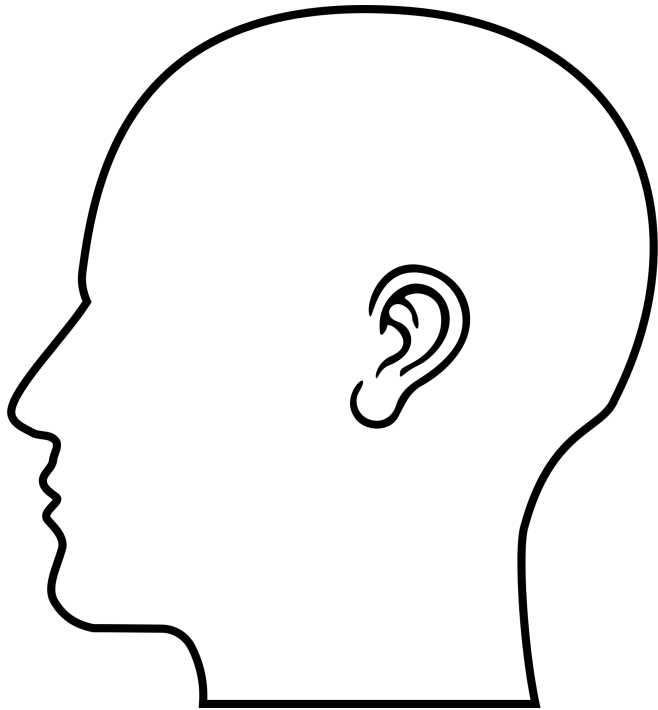
this sentence will have two phrase structures assigned to it; it can be analyzed as "they - are - flying planes" or "they - are flying - planes." And in fact, this sentence is ambiguous in just this way; we can understand it as meaning that "those specks on the horizon - are - flying planes" or "those pilots - are flying - planes."

- they - are - flying planes
- they - are flying - planes



How it works

- Parsing:
 - recognize a word from the input signal
 - look it up in LEX
 - heads go in an Initial Workspace (WS)
 - Merge fires!



[They] [are] [flying] [planes]

How it works

Workspaces (WS):

- planes fly_{θ:presp} V_{fly:θ} INFL_v V_{v:prog:pres} they
- planes flying_θ be_θ V_{be:θ:pres} INFL_v they
- *could be more ...*

[They] [are] [flying] [planes]

• Derivation:

External Merge (EM) *must* apply

θ -configuration

1. planes fly _{θ :presp} v_{fly: θ} INFL_v v_{v:prog:pres} they
2. {fly _{θ :presp}, planes} v_{fly: θ} INFL_v v_{v:prog:pres} they
3. {v_{fly: θ} , {fly _{θ :presp}, planes}} INFL_v v_{v:prog:pres} they
4. {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}} INFL_v v_{v:prog:pres}
5. {v_{v:prog:pres}, {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}}}} INFL_v
6. {INFL_v, {v_{v:prog:pres}, {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}}}}}}
7. {they, {INFL_v, {v_{v:prog:pres}, {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}}}}}}}}
8. {C, {they, {INFL_v, {v_{v:prog:pres}, {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}}}}}}}}
9. {C, {they, {INFL_v, {v_{v:prog:pres}, {they, {v_{fly: θ} , {fly _{θ :presp}, planes}}}}}}}}}}
10. they 3pl pres. be flying planes
11. they are flying planes

Merge output:
converged

FormCopy

Linear
Spellout

Internal
Merge
(IM)

[They] [are] [flying] [planes]

• Derivation:

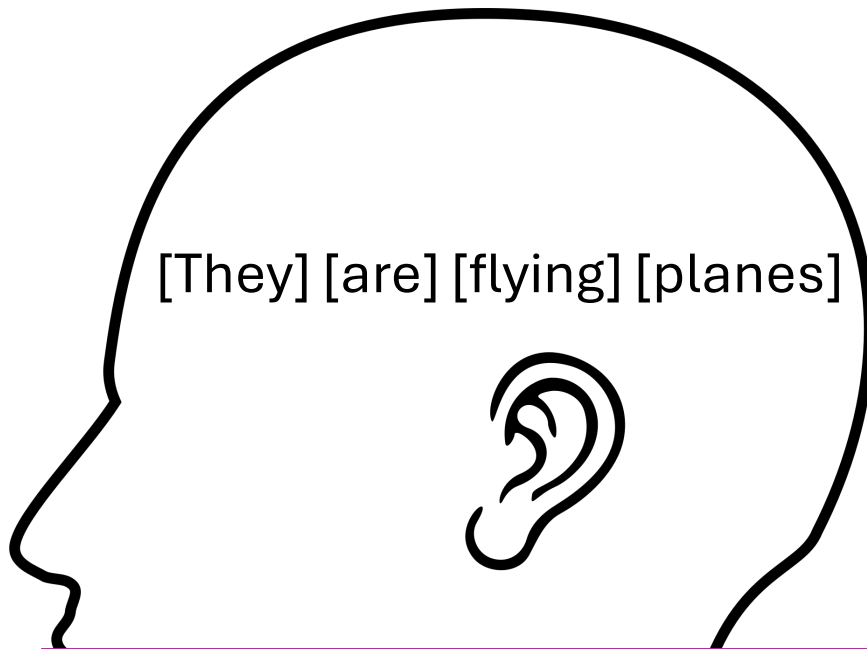
1. planes flying_θ be_θ v_{be:θ:pres} INFL_v they
2. {flying_θ, planes} be_θ v_{be:θ:pres} INFL_v they
3. {{flying_θ, planes}, planes} be_θ v_{be:θ:pres} INFL_v they
4. {be_θ, {{flying_θ, planes}, planes}} v_{be:θ:pres} INFL_v they
5. {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}} INFL_v they
6. {they, {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}}}} INFL_v
7. {INFL_v, {they, {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}}}}}
8. {they, {INFL_v, {they, {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}}}_7}}
9. {C, {they, {INFL_v, {they, {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}}}_8}}
10. {C, {they, {INFL_v, {they, {v_{be:θ:pres}, {be_θ, {{flying_θ, planes}, planes}}}_8}}

θ-configuration

Merge output:
converged

Linear
Spellout

they	3pl	pres.	be	flying	planes
they			are	flying	planes



How it works

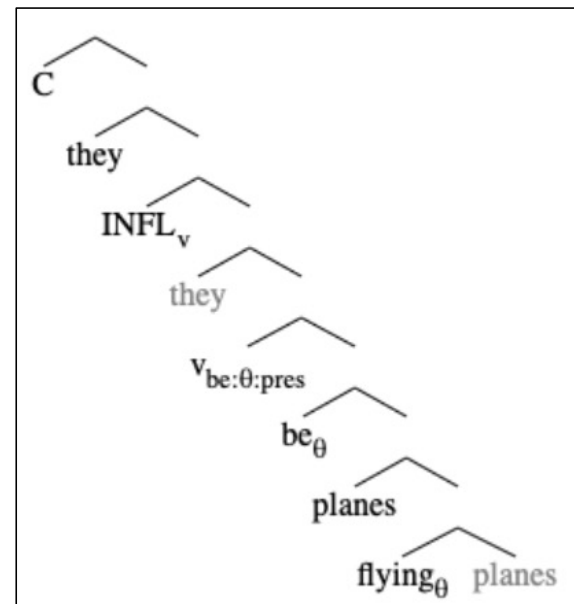
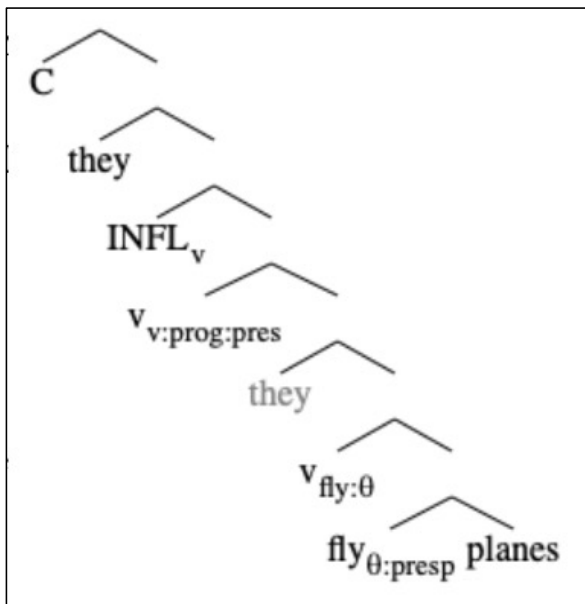
[pg.118, Chomsky (1956)]

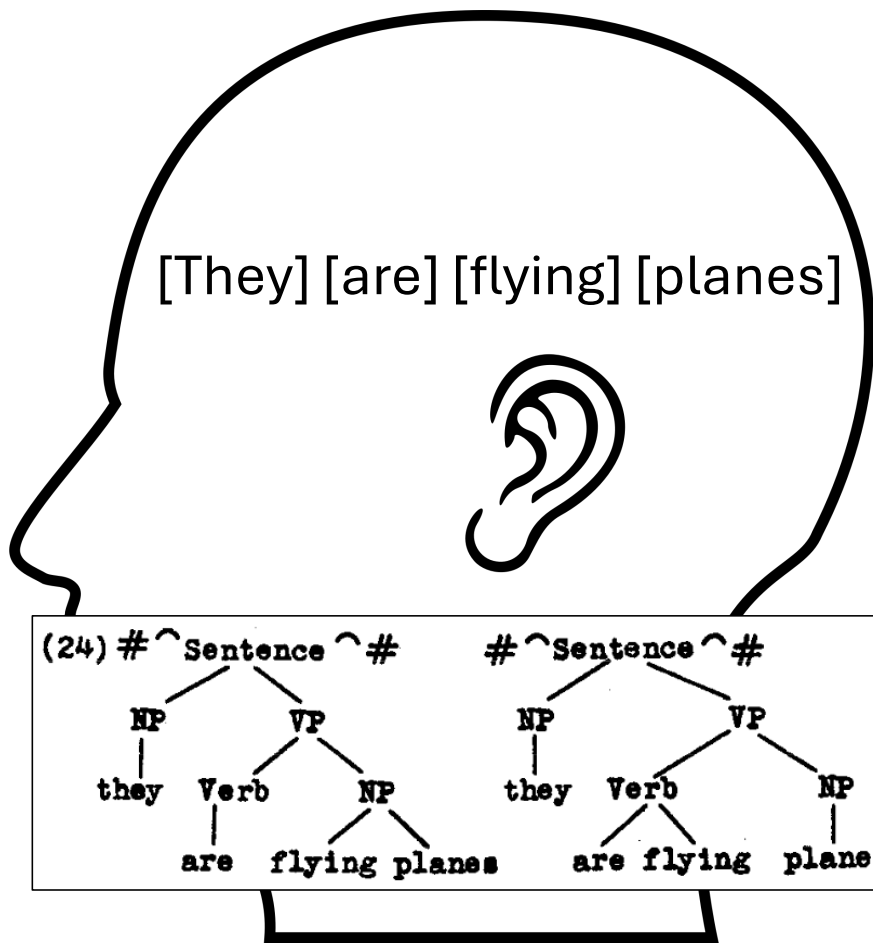
- they – are – flying planes
- they – are flying – planes

- | | | | | | | | |
|----|---|------|-----|----------|--------|--------|--------|
| 1. | {C, {they, {INFL _v , {V _{v:prog:pres} , {they, {V _{fly:θ} , {fly _{θ:presp} , planes}} ₇ | they | 3pl | pres. be | flying | planes | |
| | they | | are | flying | planes | | |
| 2. | {C, {they, {INFL _v , {they, {V _{be:θ:pres} , {be _θ , {{flying _θ , planes}}, planes}} ₈ | they | 3pl | pres. | be | flying | planes |
| | they | | are | flying | | planes | |

SMT Parser

- {C, {they, {INFL_v, {V_{v:prog:pres}, {they, {V_{fly:θ}, {fly_{θ:presp}, planes}}}}}}}
- {C, {they, {INFL_v, {they, {V_{be:θ:pres}, {be_θ, {planes, {flying_θ, planes}}}}}}}}}





How it works

[pg.118, Chomsky (1956)]

- they – are – flying planes
- they – are flying – planes

(20) $\Sigma : \# \wedge \text{Sentence} \wedge \#$
 $F: \text{Sentence} \rightarrow \text{NP} \wedge \text{VP}$
 $\text{VP} \rightarrow \text{Verb} \wedge \text{NP}$

(23) $\text{Verb} \rightarrow \text{are} \wedge \text{flying}$
 $\text{Verb} \rightarrow \text{are}$
 $\text{NP} \rightarrow \text{they}$
 $\text{NP} \rightarrow \text{planes}$
 $\text{NP} \rightarrow \text{flying} \wedge \text{planes}$

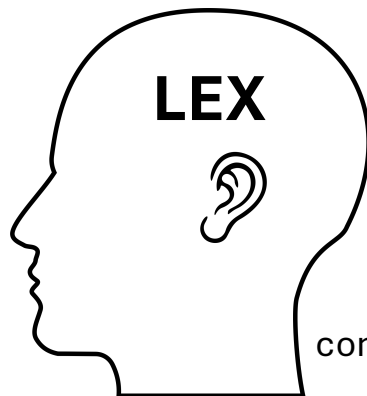
SMT Parser

sandway.arizona.edu/smtparser

Hand-built LEX

Words: they are flying planes

- ▶ **Initial WS 1:** planes fly₀:presp vfly:0 INFL_v vpred:pres INFL_v they
- ▶ **Initial WS 2:** planes flying₀ vpred:pres INFL_v they
- ▶ **Initial WS 3:** planes fly₀:presp vfly:0 INFL_v be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 4:** planes flying₀ be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 5:** planes fly₀:presp vfly:0 INFL_v vv:prog:pres they
- ▶ **Initial WS 6:** planes flying₀ vv:prog:pres they
- ▶ **Initial WS 7:** planes flying₀ vv:pass:pres they



*how many
Concepts
come to
mind?*

context, experience

WordNet LEX (nltk)

Words: they are flying planes

- ▶ **Initial WS 1:** planes flying are they
- ▶ **Initial WS 2:** plane₀ vplane:0:pres INFL_{v:3sg} flying are they
- ▶ **Initial WS 3:** planes fly₀:presp vfly:0 INFL_v are they
- ▶ **Initial WS 4:** plane₀ vplane:0:pres INFL_{v:3sg} fly₀:presp vfly:0 INFL_v are they
- ▶ **Initial WS 5:** planes flying₀ are they
- ▶ **Initial WS 6:** plane₀ vplane:0:pres INFL_{v:3sg} flying₀ are they
- ▶ **Initial WS 7:** planes flying vpred:pres INFL_v they
- ▶ **Initial WS 8:** plane₀ vplane:0:pres INFL_{v:3sg} flying vpred:pres INFL_v they
- ▶ **Initial WS 9:** planes fly₀:presp vfly:0 INFL_v vpred:pres INFL_v they
- ▶ **Initial WS 10:** plane₀ vplane:0:pres INFL_{v:3sg} fly₀:presp vfly:0 INFL_v vpred:pres INFL_v they
- ▶ **Initial WS 11:** planes flying₀ vpred:pres INFL_v they
- ▶ **Initial WS 12:** plane₀ vplane:0:pres INFL_{v:3sg} flying₀ vpred:pres INFL_v they
- ▶ **Initial WS 13:** planes flying be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 14:** plane₀ vplane:0:pres INFL_{v:3sg} flying be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 15:** planes fly₀:presp vfly:0 INFL_v be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 16:** plane₀ vplane:0:pres INFL_{v:3sg} fly₀:presp vfly:0 INFL_v be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 17:** planes flying₀ be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 18:** plane₀ vplane:0:pres INFL_{v:3sg} flying₀ be₀ vbe:0:pres INFL_v they
- ▶ **Initial WS 19:** planes flying vv:prog:pres they
- ▶ **Initial WS 20:** planes fly₀:presp vfly:0 INFL_v vv:prog:pres they
- ▶ **Initial WS 21:** plane₀ vplane:0:pres INFL_{v:3sg} fly₀:presp vfly:0 INFL_v vv:prog:pres they
- ▶ **Initial WS 22:** planes flying₀ vv:prog:pres they
- ▶ **Initial WS 23:** planes flying vv:pass:pres they
- ▶ **Initial WS 24:** planes flying₀ vv:pass:pres they

Jokes: multiple parses

- Many jokes are based on the human parser reflexively computing more than one parse
- Examples:
 - **As I handed my dad his 50th birthday card, he looked at me with tears in his eyes and said,**
 - **"You know, one would've been enough."**

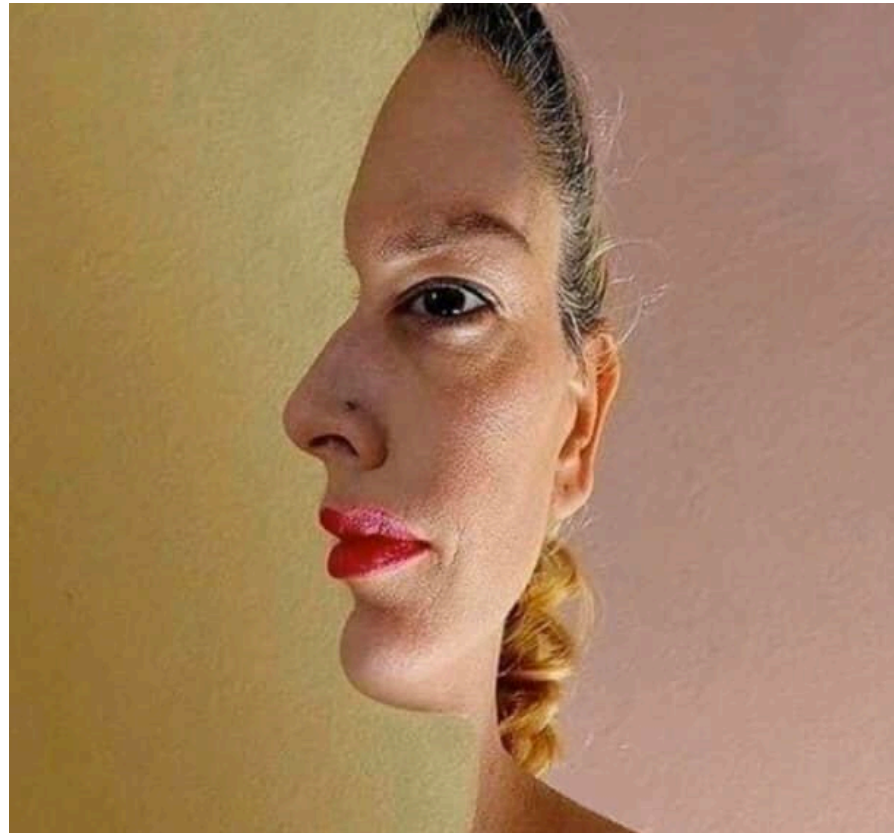
- *on a bicycle*





Pam: Is it okay to eat hot dogs with hands?
Sam: No. Hot dogs don't have hands!

Note: the visual system can also exhibit parsing ambiguity



Is it okay to eat hot dogs with hands? What's the joke here?

Yes—eating a hot dog with your hands is completely normal. In fact, it's the *default* way people eat them.

The “joke” you’re picking up on usually comes from a few places:

1. Mock “etiquette” rules

There’s a tongue-in-cheek guideline (often attributed to things like the National Hot Dog and Sausage Council) that says hot dogs should *only* be eaten with your hands—never with a knife and fork. People exaggerate this like it’s some formal dining law, which makes it funny.

2. Contrast with fancy food culture

Hot dogs are casual, messy, street food. Treating them like a fine-dining item—cutting them neatly, using utensils—creates a silly contrast. That mismatch is the joke.

3. Mild innuendo / awkward phrasing

Sometimes the humor is just in how the question is phrased (“is it okay to eat hot dogs with hands?”), as if there were some taboo or alternative method. That unexpected seriousness about something obvious can sound funny.

SA

You

Is it okay to eat hot dogs with hands? No. Hot dogs don't have hands! What's funny about this?



ChatGPT

The humor in the response "No. Hot dogs don't have hands!" comes from the unexpected twist in the interpretation of the question. Instead of addressing whether it's socially acceptable for humans to eat hot dogs with their hands, the response takes the question literally and humorously points out that hot dogs themselves don't possess hands to eat with. This unexpected and literal interpretation creates a humorous moment by subverting the original question in a clever and playful way.

Basic Property (BP) of Language

The puzzle is that from infancy on we ignore 100% of what we hear (linear order) and reflexively use only structures that we never hear but that our mind constructs, with non-trivial computations. [Chomsky GK (2021), pg.9]

- The reason must be that linear order is simply not available to the **I-language**, the system that constructs thoughts.
- Why then does speech require linearization? The articulatory system doesn't do "structure" ...
 - *The [sensory-motor] systems used for externalization have nothing to do with language; they were in place long before language emerged, and have not changed since.*
 - **Pure language is the internal system that generates thought.**

The **SMT** Parser Project

- *Isn't it then a mystery that we can parse externalized language at all?*
 - No help from **SMT** (*optimized for thought*)
 - Only **Merge** builds thoughts (**BP**)
 - Not enough time for Nature to tinker with language
 - Not enough time to evolve new systems or mechanisms
 - e.g. a parsing algorithm (CKY, Chart Parsing, Shift-Reduce, etc.)

Back to Simplicity

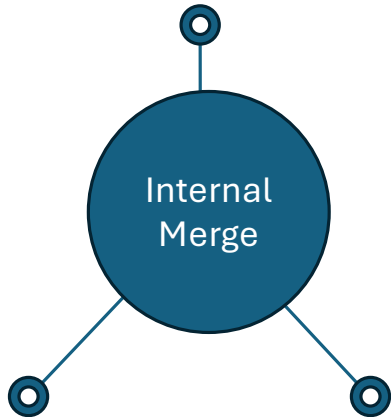
- **Deep Questions:**

- What we can do without?
- What can no longer be *state-able* in this framework?

Duality of Semantics

Earlier theory: $\{X_i, \{t_i, Y\}\}$ vs. $\{X, \{X, Y\}\}$

• $\{X, Y\}$



WS item: X has a term Y

Overlap?

$\{X, \{X, Y\}\}$

1. WS: $X Y$
2. WS': $\{X, Y\}$
3. WS'': $\{X, \{X, Y\}\}$

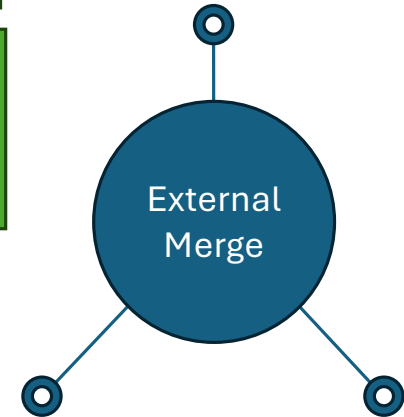
or

1. WS: $X Y X$
2. WS': $\{X, Y\} X$
3. WS'': $\{X, \{X, Y\}\}$

the copy/repetition problem

Less complex:
always preferred

• $\{X, Y\}$



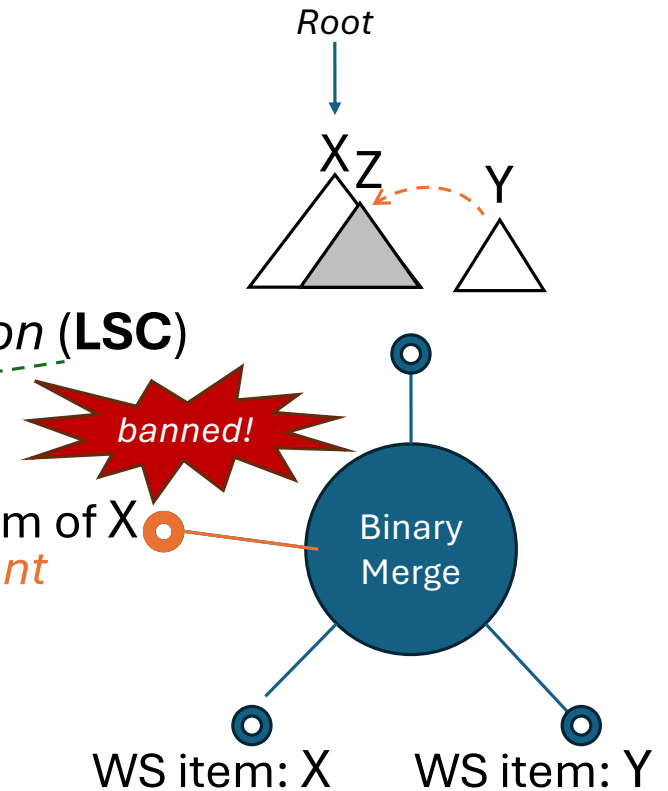
WS item: X WS item: Y

• IM for discourse/information functions

• EM for θ -configurations

Consequences

- **Merge** customized for I-Language use
 - **Duality: Language (Faculty)-Specific Condition (LSC)**
 - **Minimal Search (LSC)**
- **Design Principles:** *baked/designed-in*
 - **Extension Condition** (root-only)
 - *no tuck-in / splicing / no verbal head movement*
 - **Non-Tampering Condition (NTC)**
 - *deletion / turning something into a trace*
 - **Inclusiveness** (no invention during derivation)
 - *coindexation / γ -mark / assign a feature*
- **Merge** builds θ -configurations (*propositional domain*)
 - then transitions to the *clausal domain* via INFL_ϕ
 - then adds heads like C/C_Q, Topic, Focus etc. (*probe with language-particular spellout EXT*)



Consequences 2

- No circumvention of these principles

- e.g. **Inclusiveness** bans indexing
- work-around: insert *pre-indexed* heads, e.g. X_i X_i , into the WS

- Requires memory/marking:

- *Principle of Minimal Compliance (PMC)* (Richards 1998)

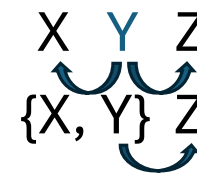
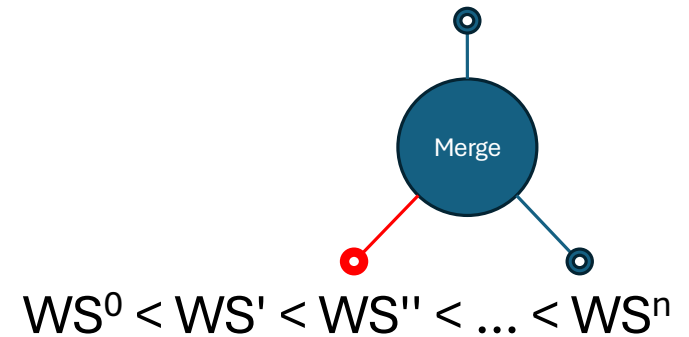
For any dependency D that obeys constraint C , any elements that are relevant for determining whether D obeys C can be ignored for the rest of the derivation for purposes of determining whether any other dependency D' obeys C .

Consequences 3

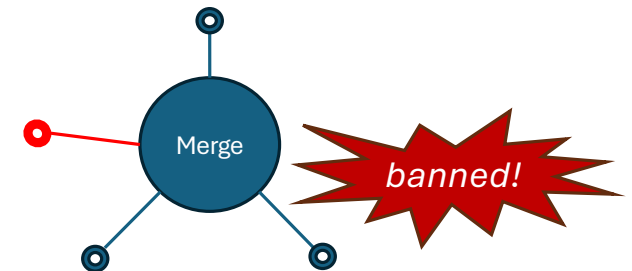
- **Idea:** check off something, it's exempt (motivation: *economy*)
- **SMT Problem:** need to mark something as exempt
 - *requires memory*
- Example (Richards 1998):
 - *Dutch: all anaphors with a clausemate binder must be zichzelf*
 - *Henk wees *zich* aan mij toe *(Henk, zich)
 - Henk wees mij aan *zichzelf* toe (Henk, zichzelf)
 - Henk wees *zich* aan *zichzelf* toe (Henk, zichzelf) mark Henk,
then ignore violation *(Henk, zich), also a timing issue
 - **Key:** *toe-wees* = 3.sg.past of *toewijzen* (assign), separable prefix *toe-*

Consequences 4

- Plain Merge means:
 - *no feature movement / inheritance*
 - *no memory devices/scratchpads*
 - *no WS history: **Markovian Assumption***
- More complex forms of Merge:
 - *useful perhaps, but also out!*
 - parallel Merge, sideways Merge
- *Minimal Yield (MY) is a **design principle***
 - *not a **filter** on Merge output*



*{X, Y} {Z, Y}
violates
Minimal Search



Markovian Assumption

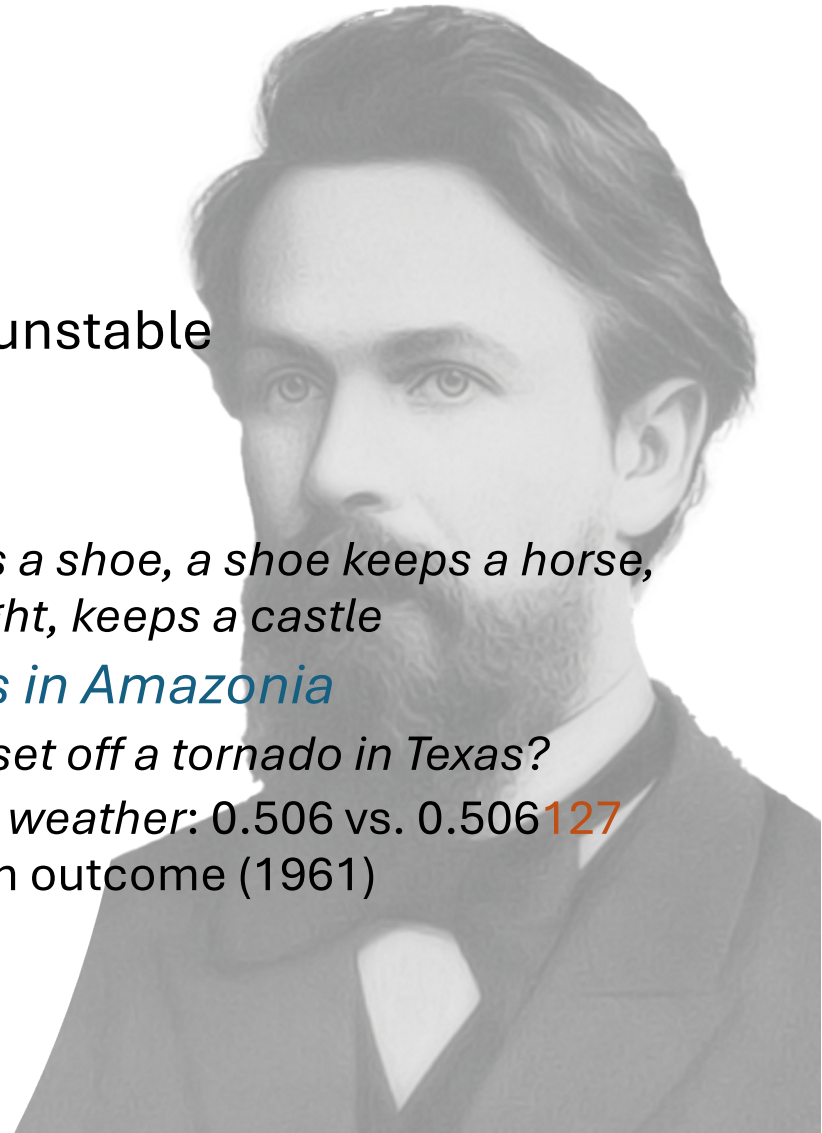
- **Independence:**
 - *each event is independent*
- **Markovian Assumption:**
 - *the future depends only on the present (not the deep past)*
- **Markov (1913), translated by Morris Halle:**
 - *Eugene Onegin*, in Russian (chapter 1 + 16 stanzas of chapter 2)
 - 20,000 characters; vowels: 8638, consonants: 11362

Probability	vowel	consonant
vowel	0.128	0.663
consonant	0.872	0.337

Strong bias in favor of alternation

Markovian Assumption

- Non-Markovian processes are complex/unstable
- Examples (Hayes 2013):
 - *kingdoms are lost for want of a nail*
 - 13th C poet: *the wise tell us that a nail keeps a shoe, a shoe keeps a horse, a horse keeps a knight, a knight, who can fight, keeps a castle*
 - *hurricanes are spawned by butterflies in Amazonia*
 - *Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?*
 - E.N. Lorenz (Chaos theory mathematician). *weather: 0.506 vs. 0.506127* resulted in a completely different simulation outcome (1961)



Minimal Search (MS)

Chomsky (2021) (3) d. carefully, the mechanic who fixed the car packed his tools

- Examples of structural relations (*all must obey MS*):
 - an adverb must **find** a verb to modify
 - *but cannot use the simplest computation: pick linearly closest verb*
 - INFL_φ must **find** θ-relevant item (*with φ for EXT*) (*Agree*)
 - θ-relevant item: *EA for transitives, IA for unaccusatives*
- Merge must also obey MS
- Formulate the **find** operation:
 - **IM** brings the *term-of* relation
 - **EM** brings the *sister-of* relation
 - **c-command** = *sister-of* + *term-of* (**Phase**-local; related: **WS** partitioning)
 - **No embellishments!**



banned!

Minimal Search (MS) 2

- **Search:** *look for something, e.g. an identical inscription*
 - in **structure:** *for free c-command* (basic: *sister-of + term-of*)
 - in **WS:** *for free member-of* (*but not inside*)
- **MS:**
 - *1st thing you find, you have to **stop*** (3rd Factor)
 - *How to know we pick A or B? Answer: you never get to see B*
- **Minimal Yield (MY):**
 - *Merge should not explode the WS search space* (Design Principle)
- **Contrast with sorting** (*finding optimal*):
 - sort requires comparisons
 - compare candidates
 - must find/generate (multiple) candidates (3rd Factor)
 - *no optimality-theoretic accounts?*



On Minimal Search (MS)

- Rules out *equidistance* (Chomsky 1995)
- Chomsky (p.c.):
 - *Right now, I don't see any reason why **any operation** should be exempt from MS.*
 - *If so, MS can include structural identity checking – which is its basic intuitive content.*

Why study the Search operation?

Because **Search** applies everywhere

- *structural dependencies* all involve search, e.g.
 - (1) an adverb must find a verb to modify,
 - (2) an anaphor an antecedent,
 - (3) EXT economy relies on *identical inscriptions*, and
 - (4) head INFL_ϕ identifies a θ -relevant item.
 - *etc.*

On Minimal Search (MS)

- **SMT:** want the simplest formulation of *Minimal Search*
- Chomsky argues there are no exceptions.
 - e.g. inputs to Merge also obey **MS**.
- *Complexity of Search:*
 - i.e. what it can find, directly affects the complexity of human language.
- **Limits of Search, limits of human language**