

The SMT Parser: where Merge and Perception meet

adapted from talk

5th International Conference on Biolinguistics and Language Acquisition

Beijing Foreign Studies University, July 12th 2026

SMT: *Strong Minimalist Thesis*

Outline:

- **Introduction:** atoms and molecules of syntax
- **SMT Parser:** system architecture
- **Theory and Computation Model:** some highlights
- **SMT:** motivating optimal simplicity

θ -configurations: *from atoms to molecules*

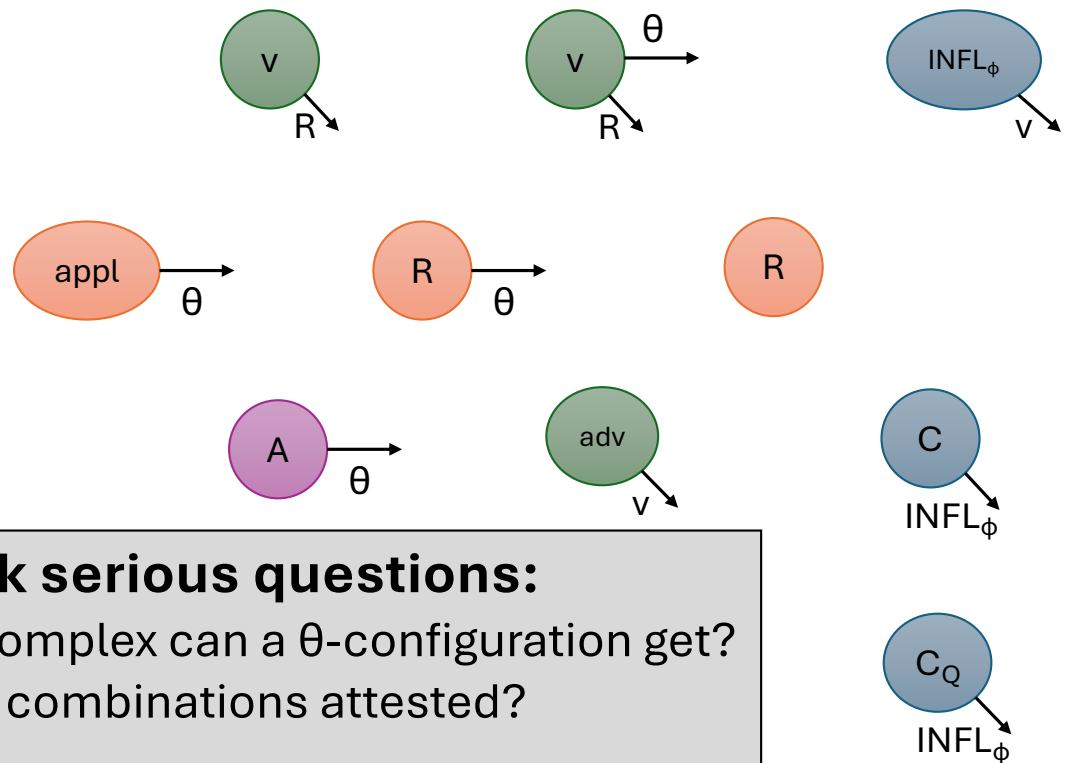
- A small inventory of *syntactic* atoms, e.g.:

- v (little v)
- R (verbal root)
- appl (Applicative)
- N (nominal) $\begin{cases} \text{deverbals} \\ A\text{-ness} \end{cases}$
- A (adjective)
- adv
- INFL _{ϕ} (inflection)
- C
- neg
- etc.

{n, R} →

A -ly →

morphemic atoms



Can ask serious questions:

- How complex can a θ -configuration get?
- Are all combinations attested?
- etc.

θ -configurations: *from atoms to molecules*

- Form molecules (**Merge**):

Minimal Search (MS)

- EXT** (Externalization) options:

- $a \text{ train}_R \text{ arrived} / * \text{ arrived}_R a \text{ train}$
- there** arrived a train (expletive)
- $un \text{ tren } llega / llega \text{ un tren}$
- $dos \text{ trenes } \textcolor{blue}{llegan} / \textcolor{blue}{llegan} \text{ dos trenes}$

- $\{v_R, \{R_\theta, IA\}\}$ ← propositional domain

- pst arrive train_a (arrived a train)

- pres llegar tren_{un} (llega un tren)

- $\{C, \{INFL_\phi, \{v_R, \{R_\theta, IA\}\}\}\}$ ← clausal domain

- 3sg pst arrive train_a

- arrived train_a

Amalgamation

IA

pronounce

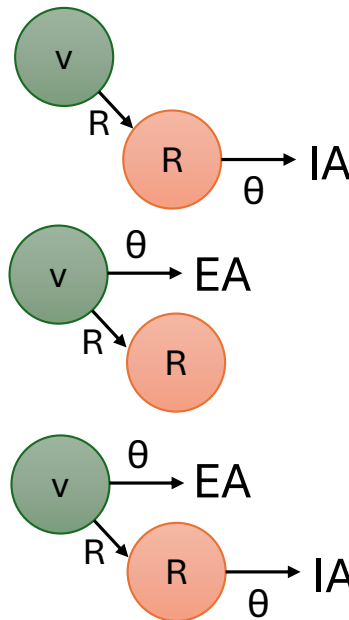
IA

don't
pronounce



θ -configurations: *standard molecules*

- There's a set of *legible* molecules wrt. **INT** (Interpretation)
 - limited set of molecules
 - information stored in **LEX** (Lexicon)
 - mapping from *words* to atoms/molecules acquired by hearer



- Unaccusative
 - $\{v_R, \{R_\theta, IA\}\}$
 - e.g. *arrive / fall*
- Unergative
 - $\{EA, \{v_{R:\theta}, R\}\}$
 - *dance / walk*
- Transitive (simple)
 - $\{EA, \{v_{R:\theta}, \{R_\theta, IA\}\}\}$
 - *like / kick*

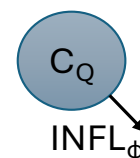
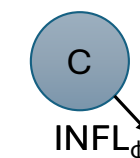
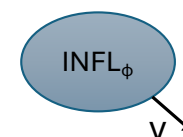
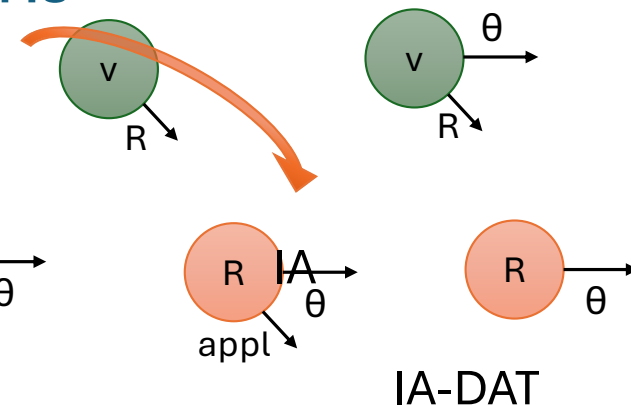


θ -configurations: *standard molecules*

- Low applicative

- e.g. *gustar* (cf. *like*)
- Él **le** gusta a ella
- "She likes him"
- $\{v_{\text{appl}}, \{IA, \{R_{\text{appl}:\theta}, \{\text{appl}_{\theta}, IA-DAT$

MS



Agreement (MS):

Me gusta el libro
me.SG.DAT please.**SG**.PRES the.**SG**.M book.**SG**.M

Me gustan los libros
me.SG.DAT please.**PL**.PRES the.**PL**.M book.**PL**.M

"I like the book/books"

θ -configurations: *standard molecules*



- *gustar* (cf. *like*)
 - $\{v_{\text{appl}}, \{IA, \{R_{\text{appl}:\theta}, \{\text{appl}_{\theta}, IA\text{-DAT}\}}\}\}$

• EXT options:

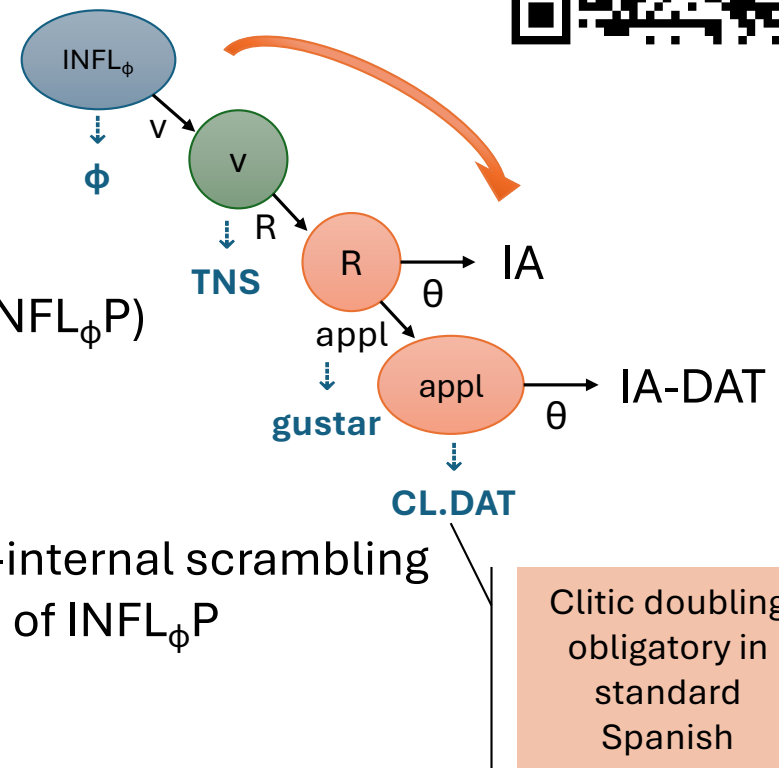
- *Él **le** gusta a ella* (IA edge of $INFL_{\phi}P$)
he CL.DAT please DAT her
"She likes him"
- ***Le** gusta él a ella*
- ***Le** gusta a ella él*
- *A ella **le** gusta él*
- ***Le** gusta él*

in situ

perhaps RP-internal scrambling

IA-DAT edge of $INFL_{\phi}P$

pro drop



θ -configurations: *standard molecules*



- *gustar* (cf. *like*)
 - $\{v_{\text{appl}}, \{IA, \{R_{\text{appl}:\theta}, \{\text{appl}_{\theta}, IA\text{-DAT}\}\}\}\}$

• **EXT** options:

- **Él gusta a ella*
- **Gusta él a ella*
- **Gusta a ella él*
- **A ella gusta él*
- **Gusta él*

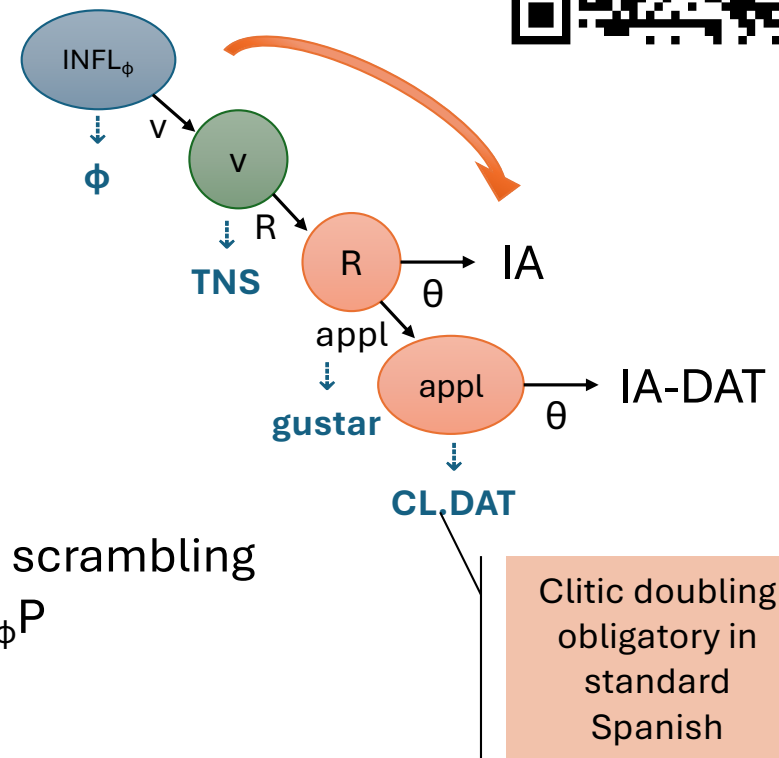
IA edge of $INFL_{\phi}P$

in situ

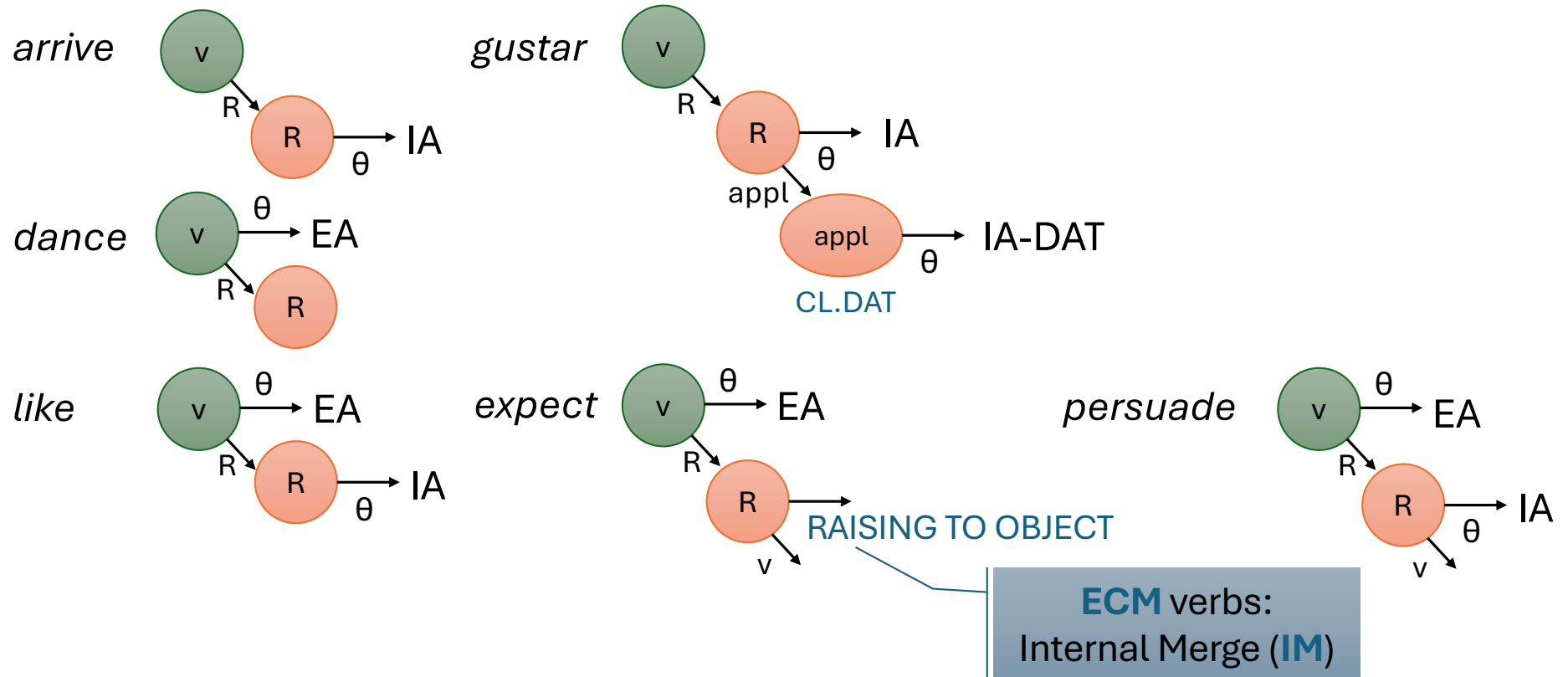
perhaps RP-internal scrambling

IA-DAT edge of $INFL_{\phi}P$

pro drop

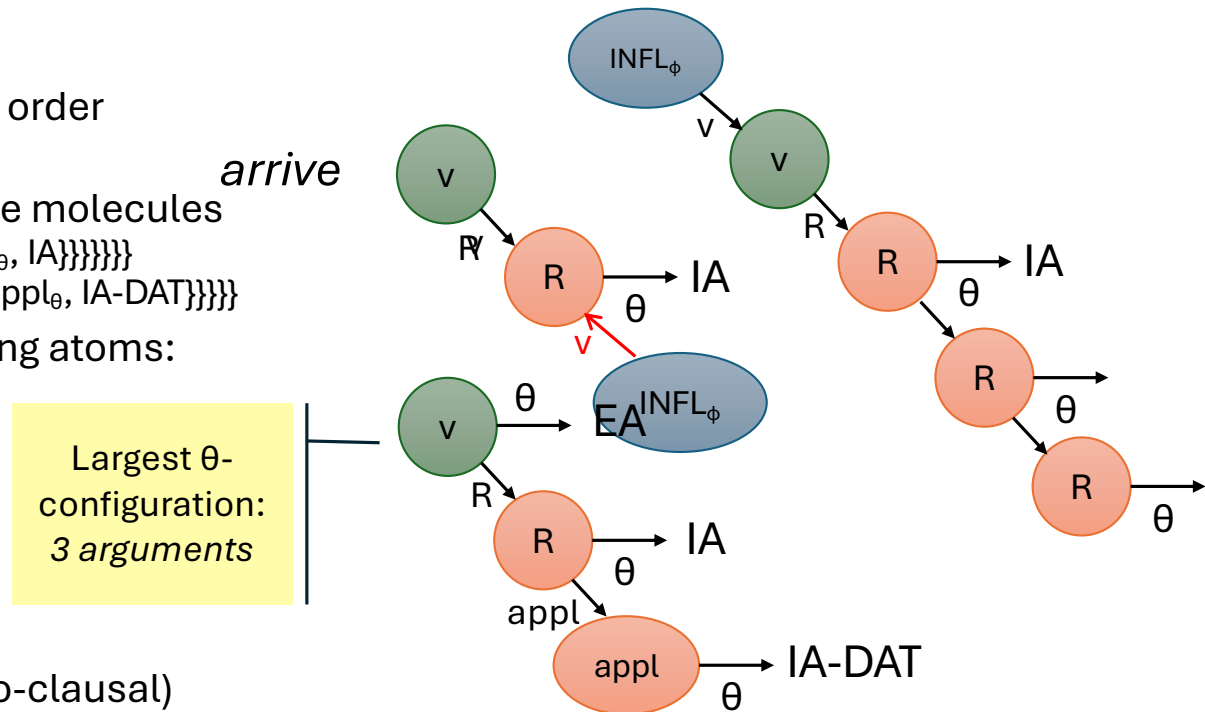


θ -configurations: *standard molecules*



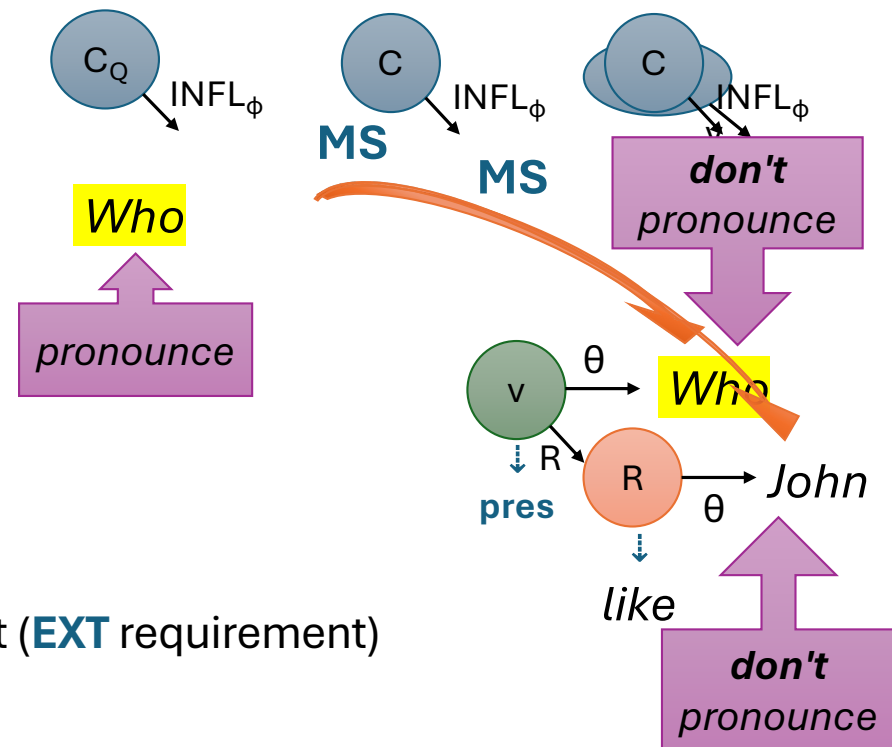
θ -configurations: some *impossible* molecules

- Can't stack in the wrong order
 - $\ast\{IA, \{R_\theta, v_R\}\}$
- Can't build arbitrary large molecules
 - $\ast\{v_R, \{IA, \{R_\theta, \{IA, \{R_\theta, \{R_\theta, IA\}\}\}\}\}\}$
 - $\{EA, \{v_{R:\theta}, \{IA, \{R_{appl:\theta}, \{appl_\theta, IA-DAT\}\}\}\}\}$
- **Note:** $\exists$ non- θ -introducing atoms:
 - *be* prog
 - *have* perf
 - *be* passive
 - *le* (了) ASP
 - *can/could* etc. modals
 - neg
 - adv / PP
- **Note:** serial verbs (mono-clausal)
 - e.g. $\{EA, \{v_{R:\theta}, \{\{R_\theta, R_\theta\}, IA\}\}\}$ or $\{EA, \{v_{R:\theta}, \{\{R_\theta, IA\}, \{R_\theta, IA\}\}\}\}$
 - argument sharing is required (*still can't violate molecule θ -limits*)
 - uses **FormSet (SBO)**
 - (*unpublished*) joint work with J. Ginsburg, H. Terada & M. Matsumoto



Clausal domain: *molecules*

- Form clausal molecules (Merge):
 - A train arrived / There arrived a train*
 - Did a train arrive?* yes-no Q
 - Did there arrive a train?*
 - What arrived?* wh-Q
 - Bill likes John*
 - $\{C, \{INFL_\phi, \{Bill, \{v_{R:\theta}, \{R_\theta, John\}\}\}\}\}$
 - Did Bill like John?* yes-no Q
 - $\{C_Q, \{INFL_\phi, \{Bill, \{v_{R:\theta}, \{R_\theta, John\}\}\}\}\}$
 - Who likes John?*
 - $\{C_Q, \{INFL_\phi, \{who, \{v_{R:\theta}, \{R_\theta, John\}\}\}\}\}$
 - Who **did** Bill like?* do-support (**EXT** requirement)
 - $\{C_Q, \{INFL_\phi, \{Bill, \{v_{R:\theta}, \{R_\theta, who\}\}\}\}\}$



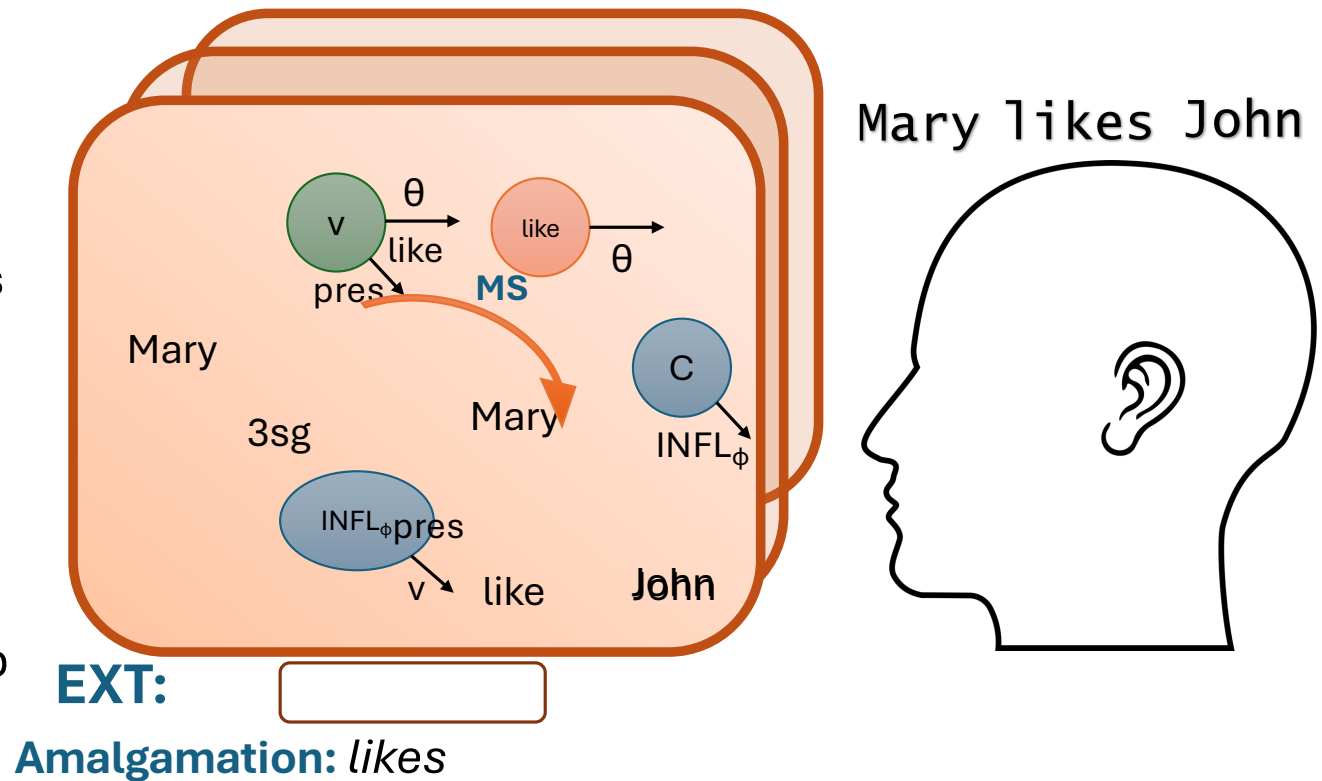
SMT: *Strong Minimalist Thesis*

Outline:

- **Introduction:** atoms and molecules of syntax
- **SMT Parser:** system architecture
- **Theory and Computation Model:** some highlights
- **SMT:** motivating optimal simplicity

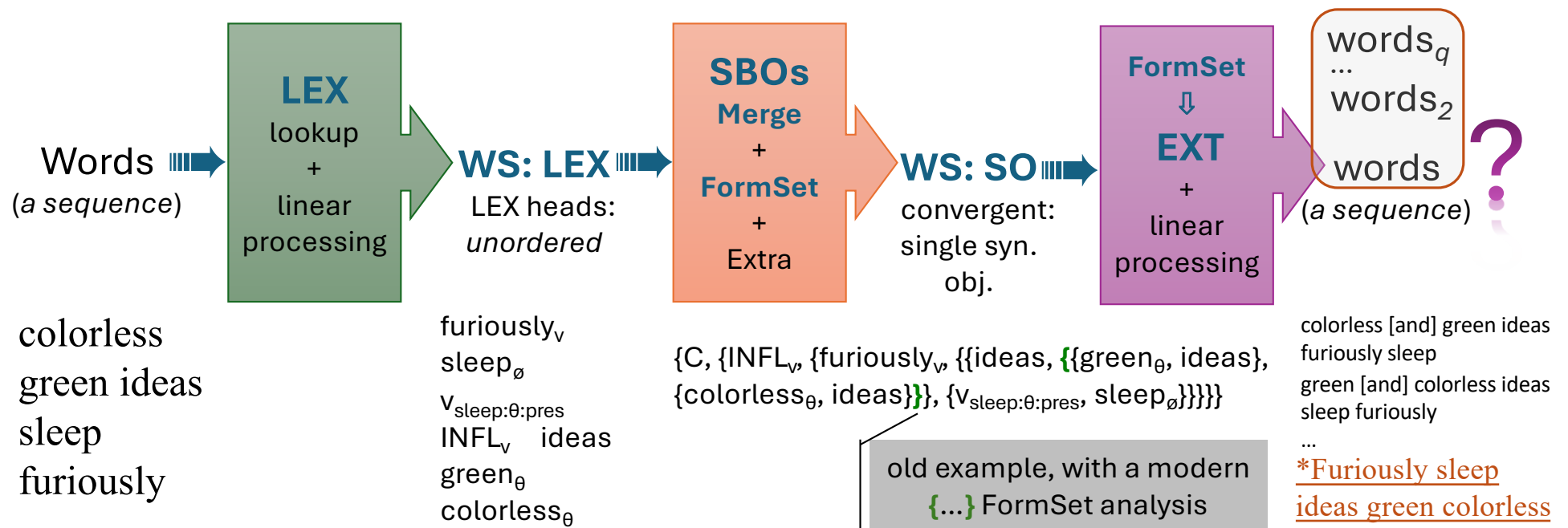
Parser Model: view #1

- Workspace (**WS**):
 - **SBO** space
 - atoms inserted from **LEX**
 - **Merge** fires! builds molecules
- **LEX** ambiguity = multiple **WS**s (*no sharing*)
 - *who*: *wh*-noun vs. relative pronoun
 - *man*: noun vs. verb



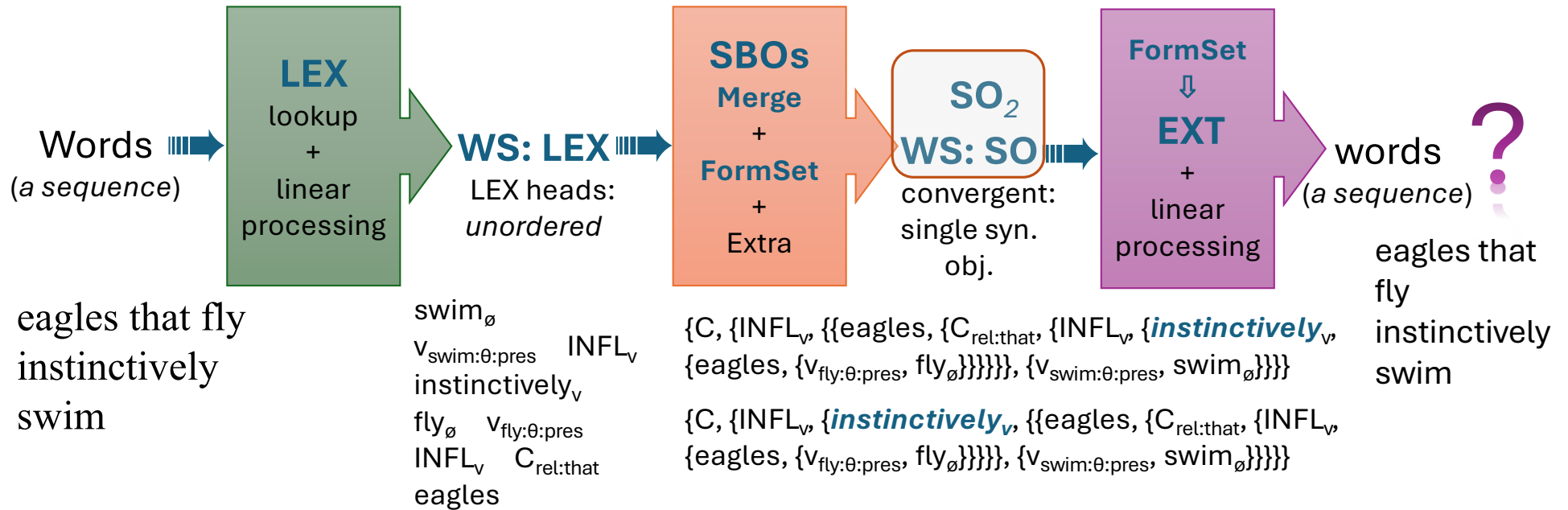
Parser Architecture: view #2

- Process view:



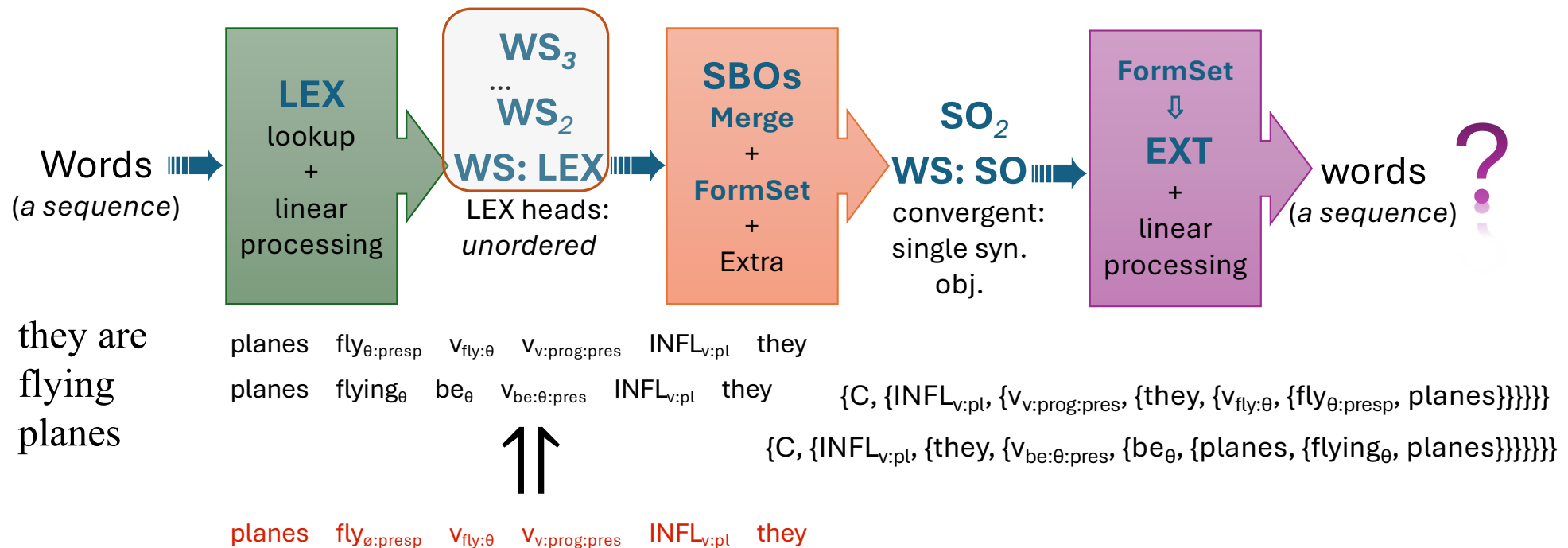
Parser Architecture: view #2

- Process view:



Parser Architecture: view #2

- Process view:



Parser Architecture: view #2

Chomsky (1955/1975), *LSLT*

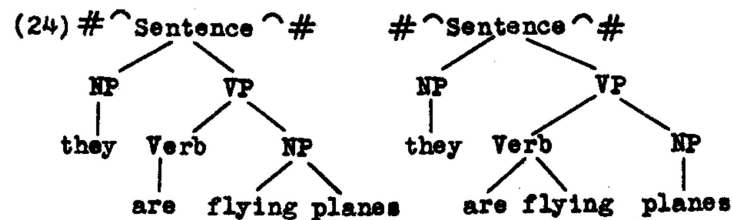
The grammar must generate a set of grammatical sentences on the basis of a limited corpus: Is it correct to identify “grammaticalness” with “significance”? I think that it is not. If we take “meaningfulness” or “significance” seriously, I think we must admit that

2 *colorless green ideas sleep furiously*

is thoroughly meaningless and nonsignificant, but it seems to me that as a speaker of English, I would regard this as a “grammatical” sentence,

Chomsky (1956), *3 models*

equivalent derivations of the sentence “they are flying planes”, reducing to the diagrams:



Hence this sentence will have two phrase structures assigned to it; it can be analyzed as

Chomsky (2013), *POP*

One early puzzle, still alive, has to do with a simple but curious fact, never recognized to be such, that it is. Consider the sentences (6):

- (6) (a) can eagles that fly swim?
(b) instinctively, eagles that fly swim

In (b), the adverb is associated with *swim*, not *fly*. Similarly, in (a), the question is about ability cases, that is clear from the interpretation, and is also shown by morphology, e.g., (7) but not (8).

- (7) are eagles that fly swimming?

- (8) *are eagles that swimming fly”

Sentence (8) does not ask whether it is the case that eagles that are swimming fly. As in the other interpretations of (6a,b), that is a fine thought, but it cannot be expressed by (8). Something about the syntax-semantics interface.



SMT: *Strong Minimalist Thesis*

Outline:

- **Introduction:** atoms and molecules of syntax
- **SMT Parser:** system architecture
- **SMT Theory and Computation Model:** some highlights
- **SMT:** motivating optimal simplicity

SMT Theory Highlights

Three examples:

- **M-gap**: subject control example
- **FormSet**: unbounded, unstructured coordination example
- **Raising to Object**: *expect* vs. *persuade*

Markovian Gap (M-gap)

Chomsky (2021), [GK](#) pg.21

Copies can be assigned a θ -role only once, but inscriptions with distinct θ -roles can be copies. That possibility is exemplified. Take the sentence form (7):

(7) John tried [X to win]

Suppose X is *John*. In this case, X can delete, unlike *John saw John*. The difference is that there is no Θ -Theory violation. There are two θ -assigners, *try* and *win*.

Application of IM to form (7) would violate DUALITY OF SEMANTICS, but there is no barrier to FC. Hence the $\langle \textit{John}, \textit{John} \rangle$ pair is an M-gap.

OF SEMANTICS. Therefore the IM-configuration must be an M-gap. In this case

M-gaps yield obligatory control — a module enabled by SMT and LSCs (specifically DUALITY OF SEMANTICS).³²

SMT therefore captures Norbert Hornstein's important insight about similarity of raising and control; similarity, but not identity, because there are two kinds of IM-configurations, one derived from Merge and the other from FC: trace and PRO in traditional terms; I'll keep the terms for expository convenience. INT cannot distinguish them, but other operations can, and do. They differ in many ways.

Markovian Gap (M-gap)

θ -configuration

- This means **perception** provides an *incomplete* **WS**
- To satisfy θ -theory, we need this SMT parser operation

Words: John tried to win

Chomsky (2021) **GK**, fn.12

¹² The most serious cases involve deletion of copies in accord with computational efficiency, leading to some of the hardest parsing problems. But there are many others.

WS 3: $\{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}$ $v_{\text{try}:\theta:\text{pst}}$ INFL_v
 WS 4: $\{v_{\text{try}:\theta:\text{pst}}, \{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}\}$ INFL_v
 WS 5: $\{\text{John}, \{v_{\text{try}:\theta:\text{pst}}, \{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}\}\}$ INFL_v
 WS 6: $\{\text{INFL}_v, \{\text{John}, \{v_{\text{try}:\theta:\text{pst}}, \{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}\}\}\}$

θ -theory: SMT parser operation

Final WS: $\{C, \{\text{INFL}_v, \{\text{John}, \{v_{\text{try}:\theta:\text{pst}}, \{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}\}\}\}\}$

INT/EXT: $\wedge \{C, \{\text{INFL}_v, \{\text{John}, \{v_{\text{try}:\theta:\text{pst}}, \{\text{try}_v, \{\text{John}, \{v_{\text{win}:\theta:\text{nonf:to}}, \text{win}_\emptyset\}\}\}\}\}\}\}$

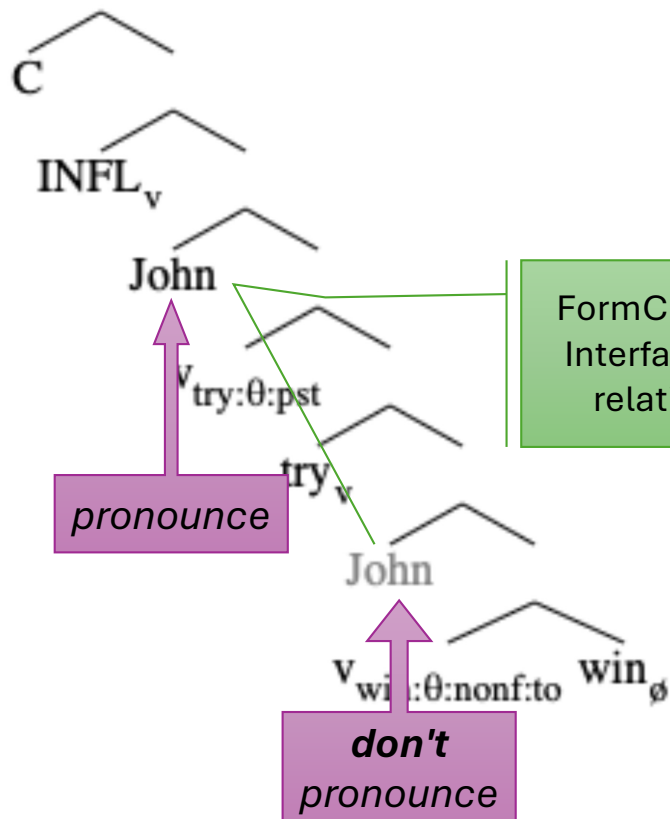
EXT: John 3sg pst try to win

Spellout: John tried to win

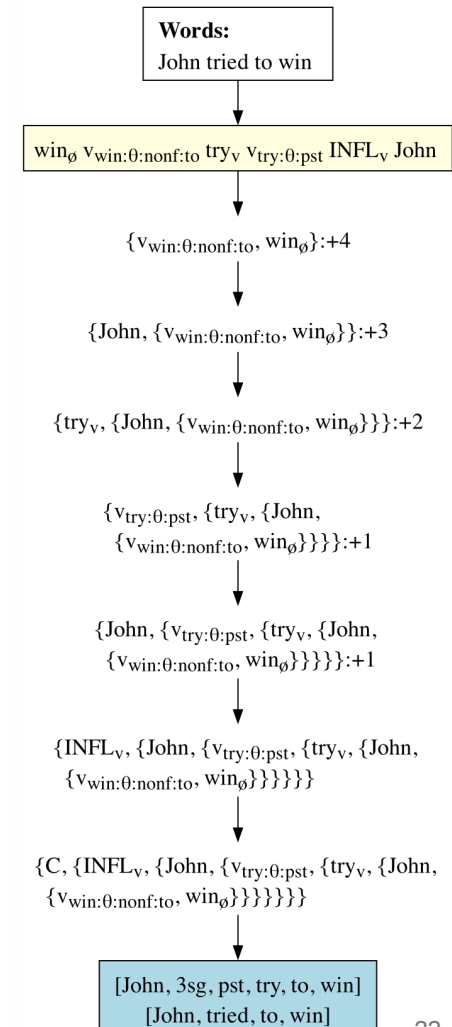
Parse found: John 3sg pst try to win

post-Merge operation
FormCopy

Markovian Gap (M-gap)



Computation:
Merge taken a step
at a time



Unbounded, Unstructured Coordination (UUS)

Chomsky (2021) [GK](#), p33

With this much background, consider one of the more complex sequences, pairing unaccusative and transitive, keeping to essentials:

(39) John arrived and met Bill.

The first step is to form the two independent objects $\{_1 v, \{arrive, John_1\}\}$ and $\{_2 John_2, \{v^*, \{meet, Bill\}\}\}$ of (40a) (*John arrived* and *John met Bill*) in the normal way, satisfying Θ -Theory. The next step is to construct the *set* (40b):

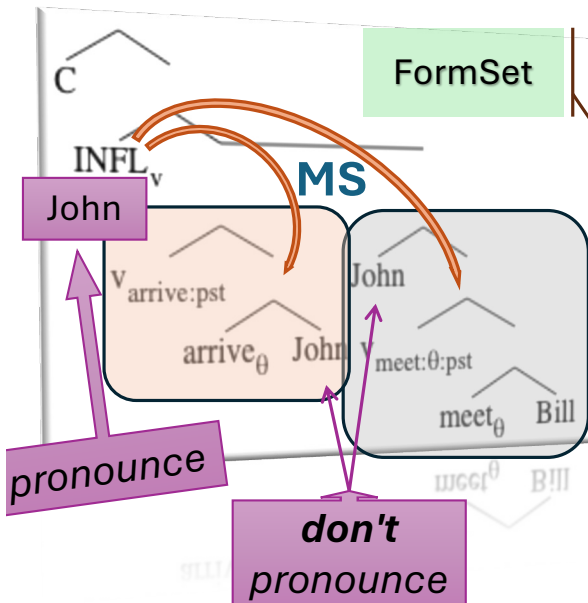
- (40) a. $\{_1 v, \{arrive, John_1\}\}, \{_2 John_2, \{v^*, \{meet, Bill\}\}\}$
 b. $\{\{_1 v, \{arrive, John_1\}\}, \{_2 John_2, \{v^*, \{meet, Bill\}\}\}\}$

(43) $C, \{John_3, \{INFL, \langle \&, \{_1 v, \{arrive, John_1\}\}, \{_2 John_2, \{v^*, \{meet, Bill\}\}\} \rangle\}$

Either $John_1$ or $John_2$ can raise by IM to the position of $John_3$. The one that does not raise is determined by FC to be a copy of $John_3$. INT cannot determine which one raised. But it doesn't matter. The choice doesn't affect later operations: there is no indeterminacy. There is also no ambiguity: the interfaces are not affected. Whichever choice is taken, what reaches CI and SM is the same.

Unbounded, Unstructured Coordination (UUS)

θ -theory: SMT parser operation



• Initial WS: an incomplete reconstruction

Words: John arrived and met Bill

▼ Initial WS 1: Bill meet_θ v_{meet:θ:pst} INFL_v arrive_θ v_{arrive:pst} INFL_v John

▼ WS 1: {meet_θ, Bill} v_{meet:θ:pst} INFL_v arrive_θ v_{arrive:pst} INFL_v John

WS 2: {v_{meet:θ:pst}, {meet_θ, Bill}} INFL_v arrive_θ v_{arrive:pst} INFL_v John [gray]

WS 3: {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}} INFL_v arrive_θ v_{arrive:pst} INFL_v

WS 4: {arrive_θ, John} {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}} INFL_v v_{arrive:pst} INFL_v

WS 5: {v_{arrive:pst}, {arrive_θ, John}} {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}} INFL_v INFL_v

WS 6: {{v_{arrive:pst}, {arrive_θ, John}}, {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}} INFL_v INFL_v

WS 7: {INFL_v, {{v_{arrive:pst}, {arrive_θ, John}}, {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}}}}

Final WS: {C, {INFL_v, {{v_{arrive:pst}, {arrive_θ, John}}, {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}}}}

INT/EXT: $\wedge$ {C, {INFL_v, {{v_{arrive:pst}, {arrive_θ, John}}, {John, {v_{meet:θ:pst}, {meet_θ, Bill}}}}}}

EXT: John {[3sg,pst,arrive],[and],[3sg,pst,meet,Bill]}

Spellout: John {[arrived],[and],[met,Bill]}

Parse found: John {[3sg,pst,arrive],[and],[3sg,pst,meet,Bill]}



Unbounded, Unstructured Coordination (**UUS**)

- Incomplete reconstruction of the **WS**:
 - has desirable **3rd Factor** consequences
 - Example: $[_{NP}$ *The mechanic who fixed the car*] arrived and met Bill
 - θ -theory: build the complex NP twice and **EM** twice
 - θ -theory **SMT parser operation** only builds it once!

θ -configuration

Words: the mechanic who fixed the car arrived and met Bill

Initial WS 1: Bill meet _{θ} v_{meet:0:pst} INFL_v arrive _{θ} v_{arrive:pst} INFL_v car_{the} fix _{θ} v_{fix:0:pst} INFL_v C_{rel:who} mechanic_{the}

...

WS 7: { {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}}, {v_{meet:0:pst}, {meet _{θ} , Bill}} } INFL_v arrive _{θ} v_{arrive:pst} INFL_v

WS 8: {arrive _{θ} , {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}} } { {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}} } {v_{meet:0:pst}, {meet _{θ} , Bill}} } INFL_v v_{arrive:pst} INFL_v

...

Final WS: {C, {INFL_v, { {v_{arrive:pst}, {arrive _{θ} , {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}} } {v_{meet:0:pst}, {meet _{θ} , Bill}} } }

INT/EXT: {C, {INFL_v, { {v_{arrive:pst}, {arrive _{θ} , {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}} } { {mechanic_{the}, {C_{rel:who}, {INFL_v, {mechanic_{the}, {v_{fix:0:pst}, {fix _{θ} , car_{the}}}}}}} } {v_{meet:0:pst}, {meet _{θ} , Bill}} } }

EXT: the mechanic who 3sg pst fix the car {[3sg,pst,arrive],[and],[3sg,pst,meet,Bill]}

Spellout: the mechanic who fixed the car {[arrived],[and],[met,Bill]}

Parse found: the mechanic who 3sg pst fix the car {[3sg,pst,arrive],[and],[3sg,pst,meet,Bill]}

θ -theory: SMT parser operation

Unbounded, Unstructured Coordination (UUS)

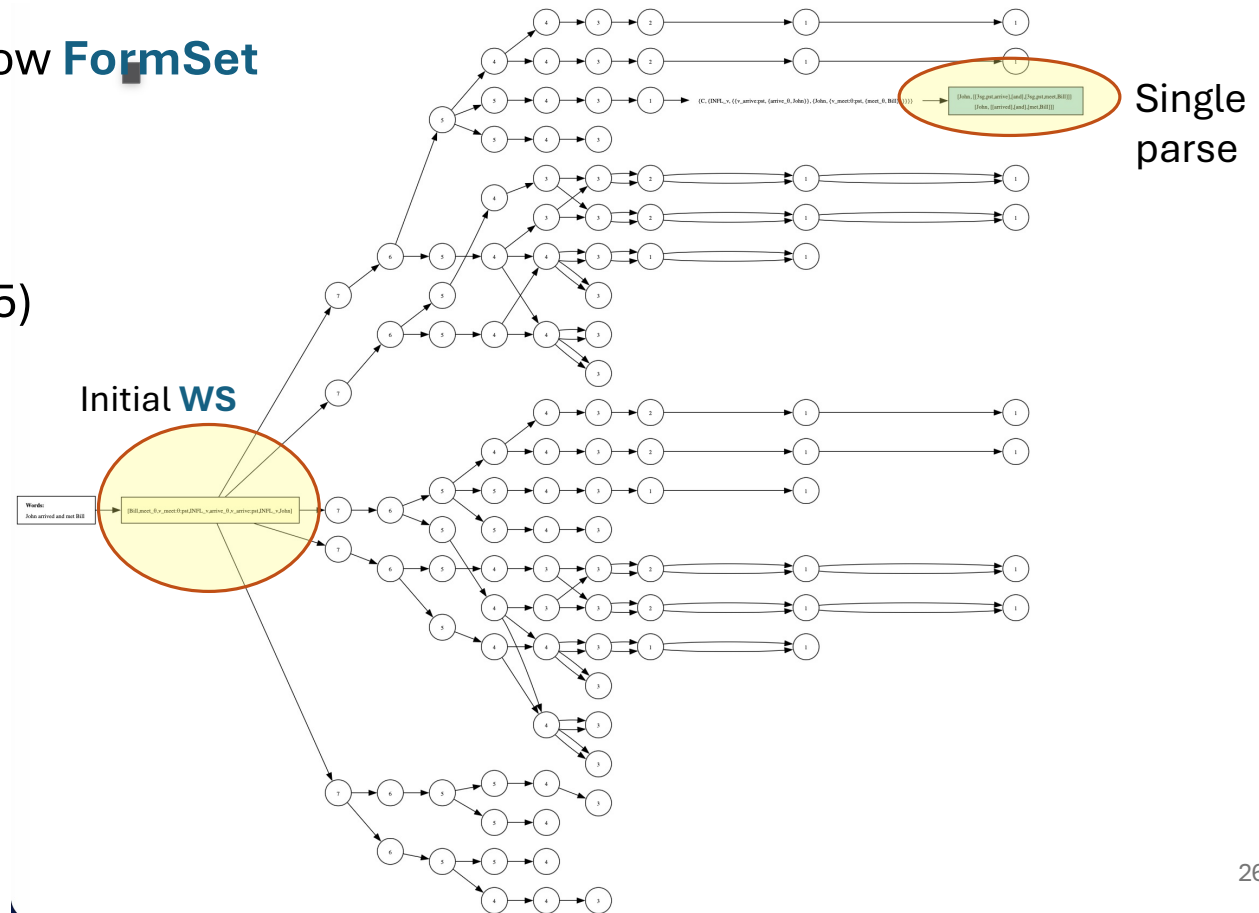
For more detail on how **FormSet** works, see:

- Chomsky (2021)
- Chomsky (2024)
- Fong & Oishi (2025)

Computation space:

represented here as a graph

- vertex = **WS** state
- edge = **Merge** step



Raising to Object: *expect* vs. *persuade*

Chomsky (2021), **GK**, p24

So far we have been considering raising to subject. Let's turn to raising to object, which should work much the same way:

- (15) a. John expected Bill to leave
b. John persuaded Bill to leave

These should be analogous to the subject-raising/control cases already discussed.

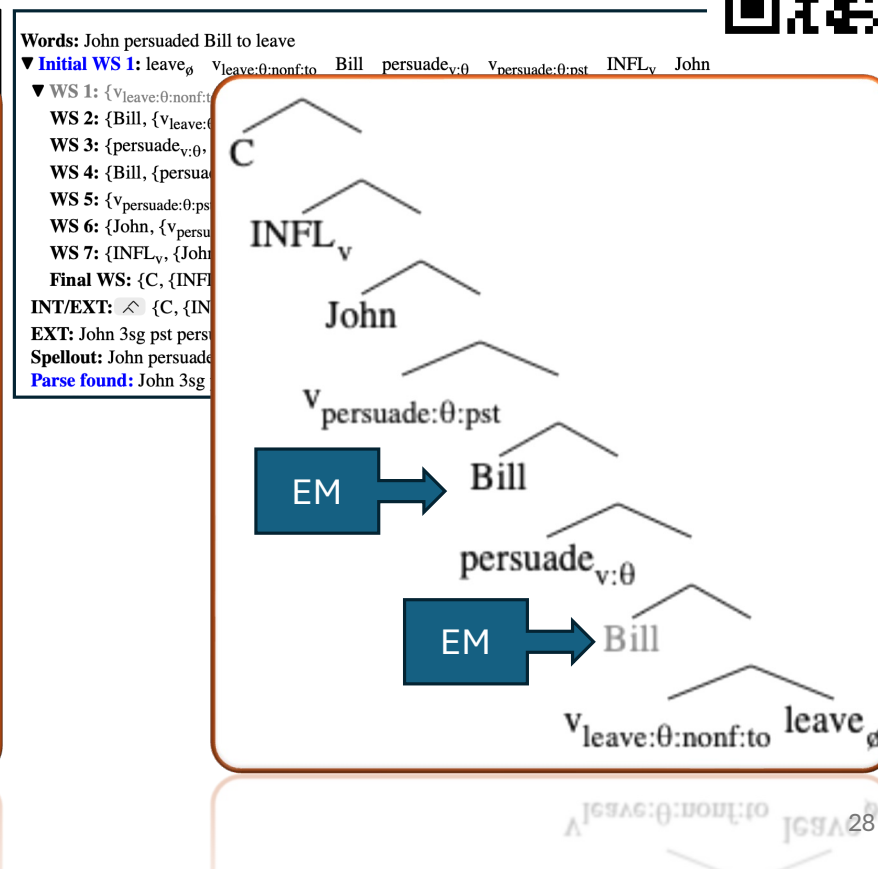
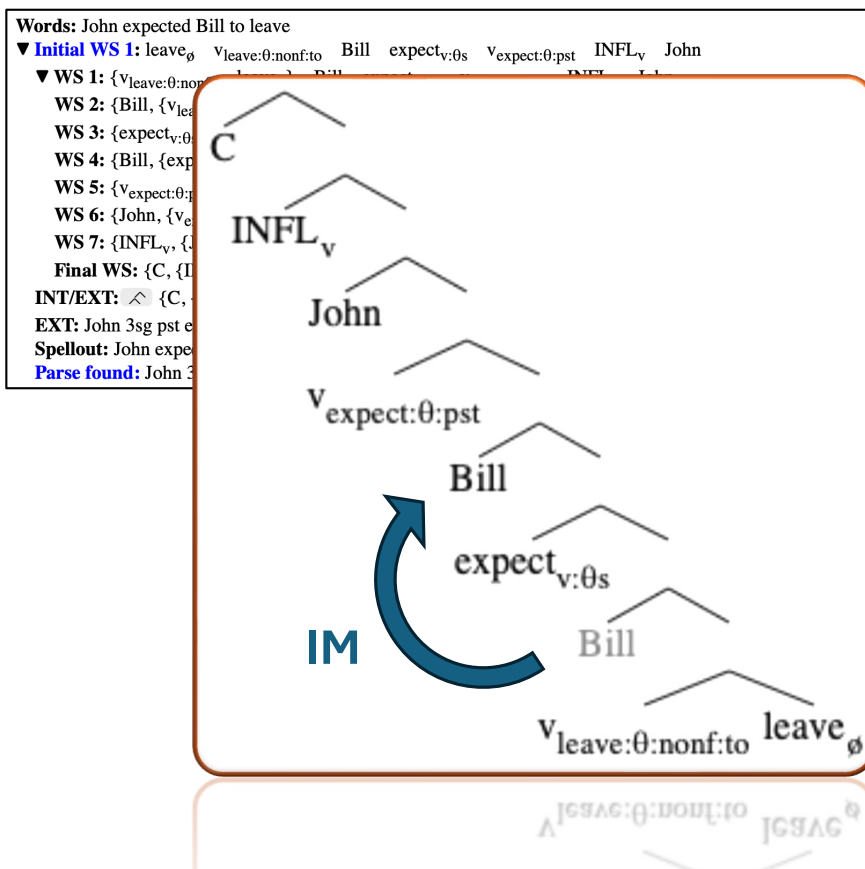
In (15a), *Bill* raises by IM to SPEC-*expect*. In (15b), *Bill* is merged by EM with the complex verbal structure {*persuade*, {*Bill to leave*}} as its direct object. Within our highly restricted framework, the structures should therefore be as in [H] skipping details, where V is either *expect* or *persuade*:

[H] John INFL {₂ Bill₁, {₁ V, {Bill₂ to leave}}}

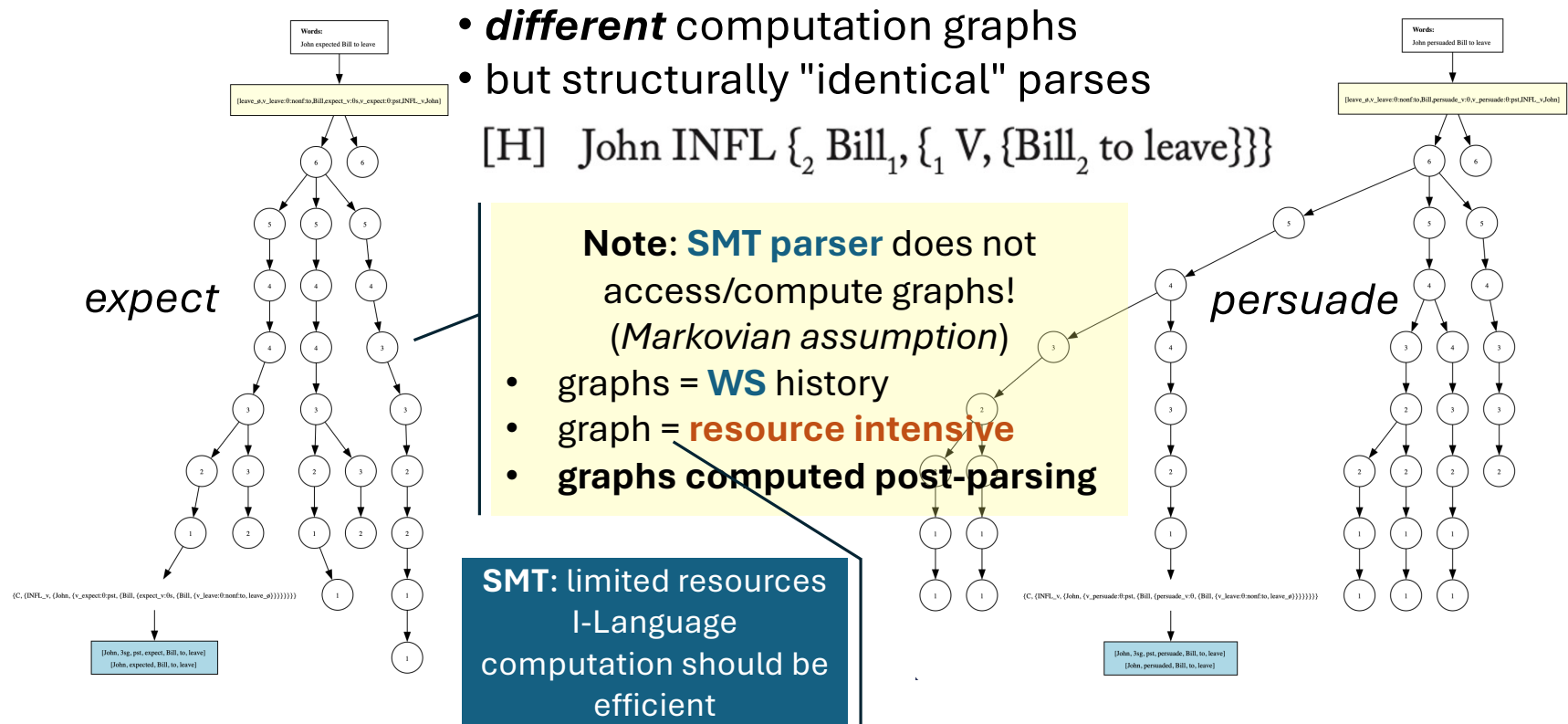
In both cases, FC determines that the two occurrences of *Bill* are copies. If V = *expect*, the paired Copies were generated by IM; if V = *persuade*, *Bill*₁ was generated by EM, with its own θ -role; it is an M-gap, a PRO structure. In both cases MS determines deletion of *Bill*₂. Thus object control falls into place along with subject control, crucially relying on the distinction between the reincarnation of trace and PRO, now within SMT.



Raising to Object: *expect* vs. *persuade*



Steps (*to the parse*): computation graphs



SMT: *Strong Minimalist Thesis*

Outline:

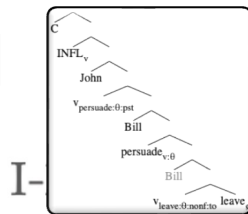
- **Introduction:** atoms and molecules of syntax
- **SMT Parser:** system architecture
- **Theory and Computation Model:** some highlights
- **SMT:** motivating optimal simplicity

SMT: Strong Minimalist Thesis

- **SMT Parser:** respect fundamental constraints

stemming:

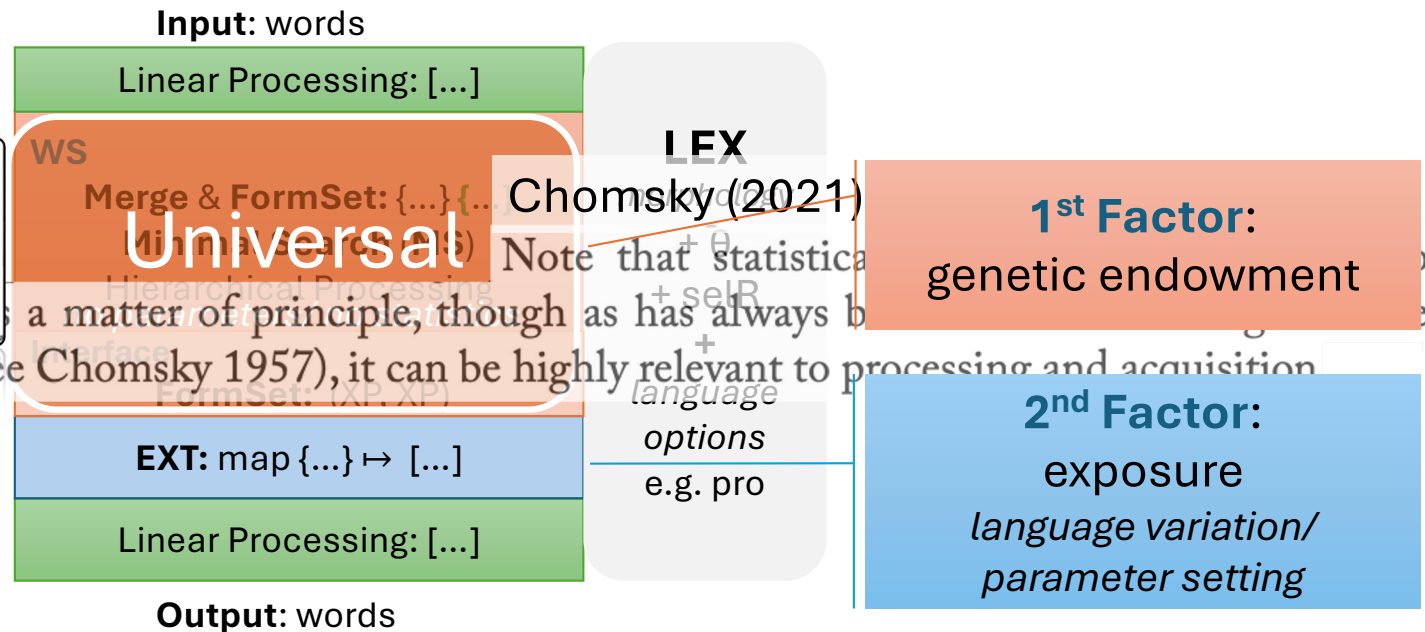
unpack inflected verb/noun/adj



I-
enterprise (see Chomsky 1957), it can be highly relevant to processing and acquisition

affixation:

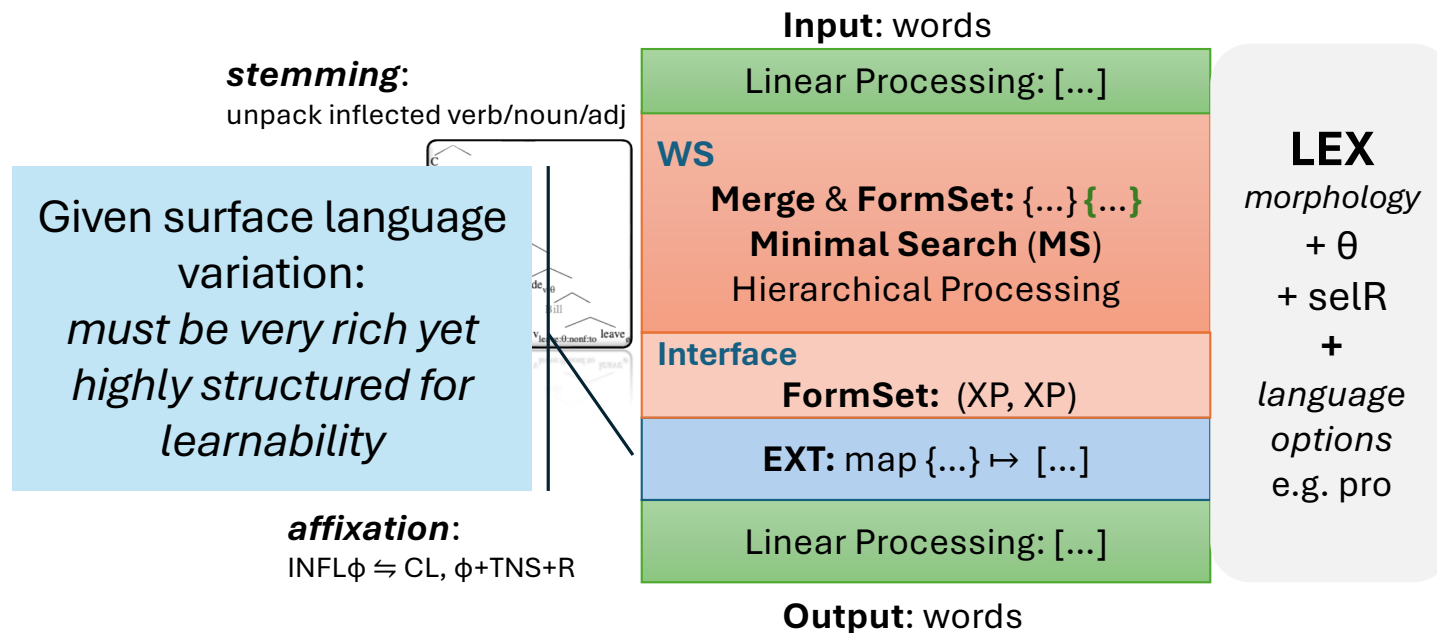
$\text{INFL}\phi \rightleftharpoons \text{CL}, \phi + \text{TNS} + \text{R}$



SMT: Strong Minimalist Thesis

Chomsky (2021) [GK](#), fn.14

¹⁴ Externalization should not be confused with processing. It is pure syntax,





WS Complexity

Wor Chomsky (2021) GK, p9

- **Ini** (3) a. the mechanic fixed the car carefully
- **Ini** b. the mechanic carefully packed his tools
- c. the mechanic who fixed the car carefully packed his tools

fully packed his tools

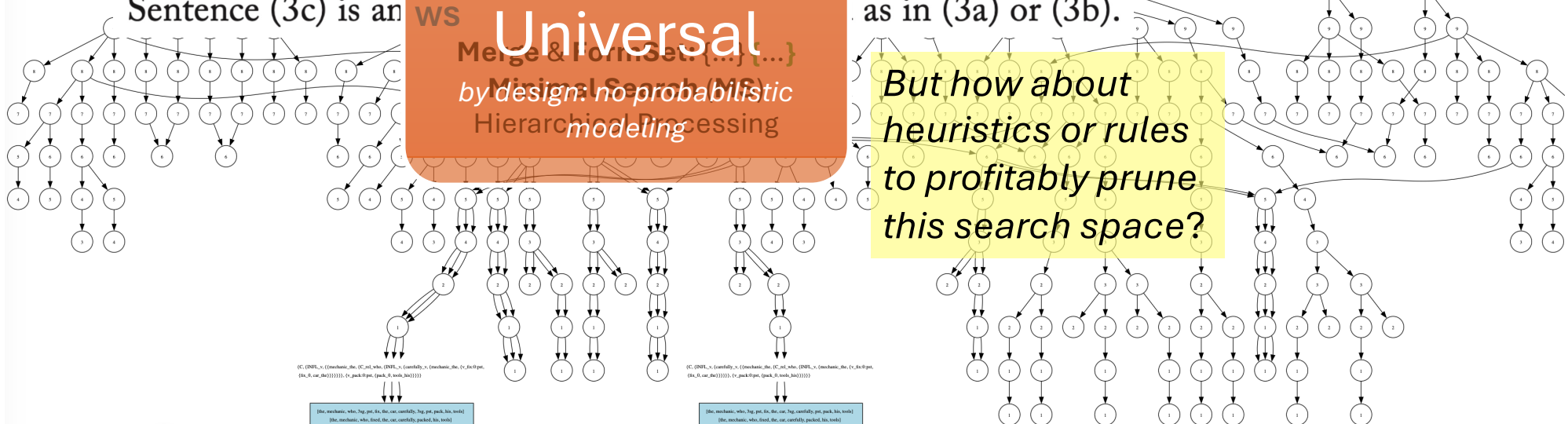
INFL_v who mechanic_{the}
INFL_v C_{rel:who} mechanic_{the}

Sentence (3c) is an

Universal
Merge & FormSet {...} {...}
Minimal Search (MS)
Hierarchical Processing

as in (3a) or (3b).

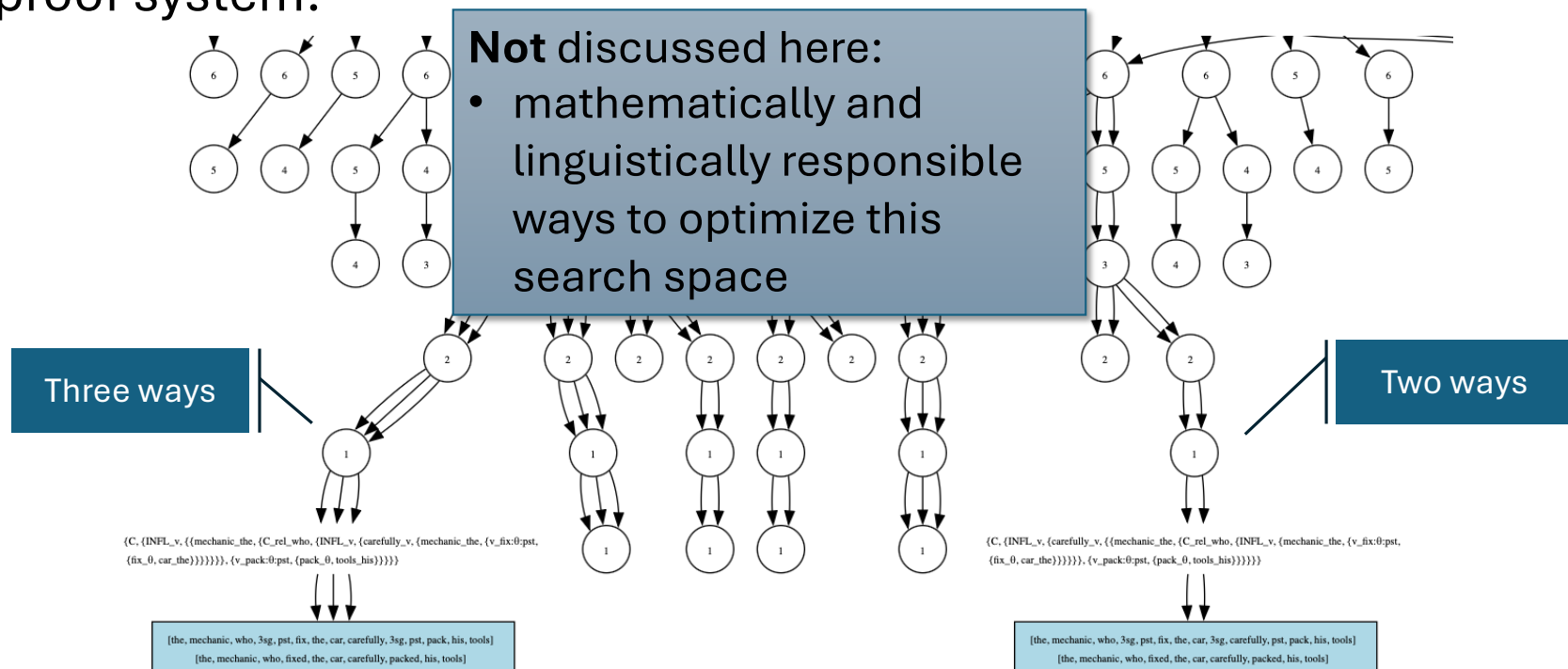
*But how about
heuristics or rules
to profitably prune
this search space?*



WS Complexity

†only reason: *debugging is easier*

- **SMT Parser:** in serial mode[†] (not *parallel* mode), a (linearized) proof system:



WS Complexity

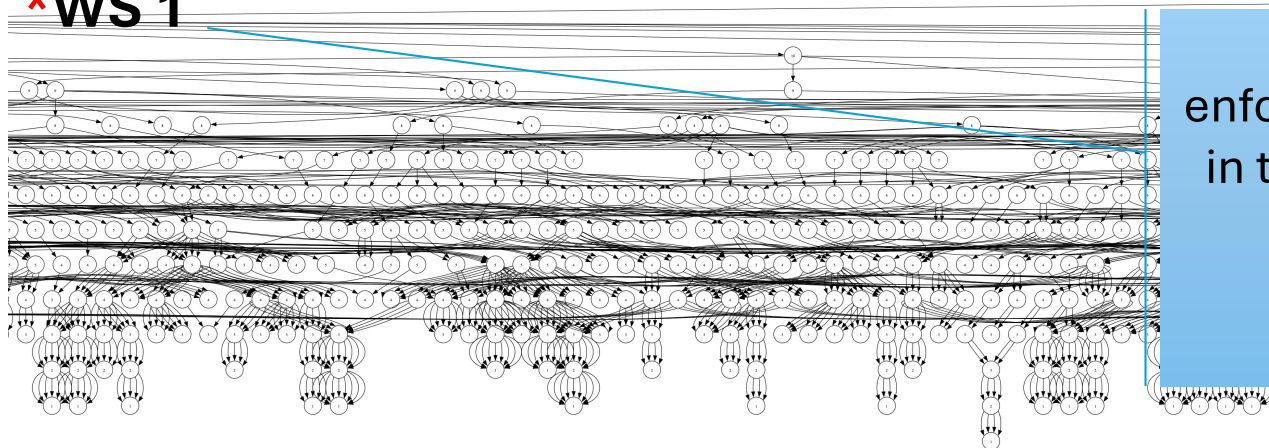
Words: the mechanic who fixed the car carefully packed his tools

► **Initial WS 1:** tools_{his} pack_θ v_{pack:θ:pst} INFL_v carefully_v car_{the} fix_θ v_{fix:θ:pst} INFL_v who mechanic_{the}

► **Initial WS 2:** tools_{his} pack_θ v_{pack:θ:pst} INFL_v carefully_v car_{the} fix_θ v_{fix:θ:pst} INFL_v C_{rel:who} mechanic_{the}

- two **WSs**, only one converges, **WS 2's** computation graph shown earlier
 - **WSs** are silo-ed, assume they proceed in parallel (**Merge** from one cannot access the other)
- **LEX access** subject to statistics / context relevance, e.g. *man* as a noun vs. verb; **surprisal**

*WS 1



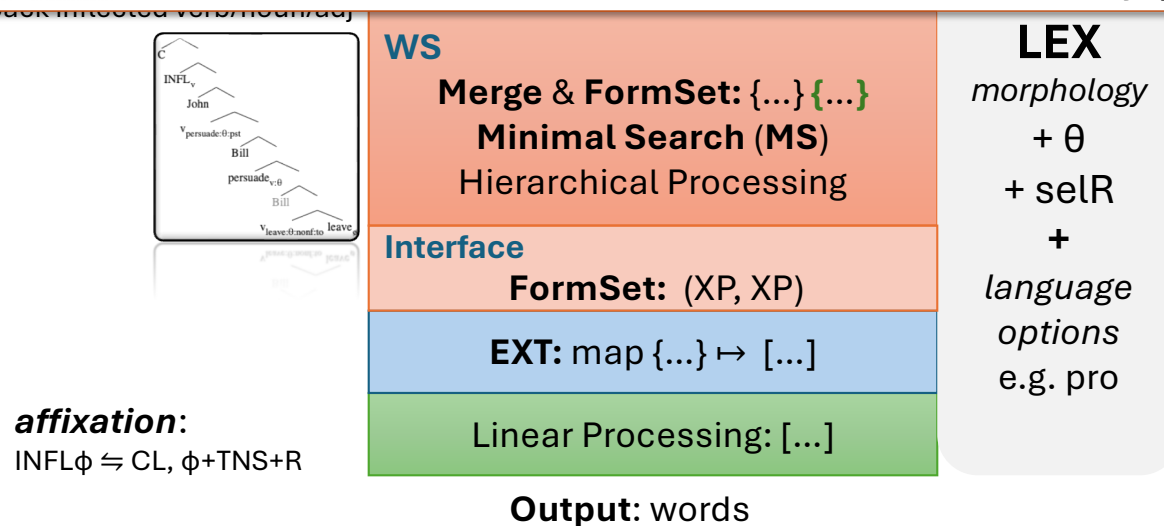
3rd Factor:
enforce efficient computation
in the combinatorial system
(of **SMT**):
"usable" language
resource restrictions

SMT: Strong Minimalist Thesis

- Robustness of **SBO** despite lack of exposure or level of surprisal

It is easy to fall into error on these matters, as I did in *LSLT* (Chomsky 1955) in seeking to determine word boundaries in I-language in terms of transitional probabilities, part of the failure in those early days to distinguish sharply between explanation and procedural analysis of data.

Chomsky (2021) **GK**, fn.14



SMT = Ultimate Simplicity

- Reason for (*forced*) simplicity of operations:
 - evolutionary time-scale ($\approx 300\text{kya}$ is brief): **I-Language** coinciding with the appearance of anatomically-modern humans.

Returning to structure-dependence, the problem of learnability is overcome if the property is part of UG, as seems clearly to be the case. There is, then, no learning. The problem of evolvability is overcome if the property follows from the simplest combinatorial operation, one that must exist if language is to exist at all.¹⁷ In fact, it does. The simplest combinatorial operation is binary set-formation, Merge in contemporary terminology — though the matter is more intricate as we see directly. Structure-dependence follows at once. Linear order is not an option for the child if computation is based on the simplest operation, the one yielded by evolution in its normal course.

SMT: Strong Minimalist Thesis

- **Merge** is fundamentally **bottom-up** (*Nature doesn't care!*)
- The **SMT Parser** eschews all parsing algorithms and mechanisms,

Simplest possible:

WS: $\dots, X, \dots, Y, \dots \mapsto \text{WS}': \dots, \{X, Y\}, \dots$ (**EM**)

WS: $\dots, XP, \dots \mapsto \text{WS}': \dots, \{Y, XP\}, \dots$ (**IM**)

XP **MS** contains Y No sideways, no parallel Merge, etc.

Must adopt this!

*won't have simultaneously evolved two distinct **SBOs**, one for thought and one for parsing*

structure-dependence, relies on elementary properties of computational efficiency: adopting the simplest computational operation and the No-Tampering Condition (NTC), which basically says “don’t add bells and whistles but keep to pure elegance.”

$\text{INFL}\phi \Rightarrow \text{CL}, \phi + \text{INS} + \text{R}$

Output: words

SMT: Strong Minimalist Thesis

- **Merge** is fundamentally bottom-up
- Minimal Search (**MS** or Σ):

*find 1st matching element and stop (no comparisons/**equidistance**)*

3rd Factor:

ultimate simplicity also
= most efficient
limits "**usable**" language

Another natural condition is limiting search. For an operation O to apply to items it must first locate them. It must incorporate an operation Σ that searches LEX and WS and selects items to which O will apply. It is fair to take Σ to be a third factor element, on the shelf and available for any operation.²⁷

We assume here that minimization of search is Nature's guide in seeking simple solutions for computational systems, the reconstruction stage of evolution discussed above.

$\text{INFL} \rightarrow \text{CE}, \text{PTINSTN}$

Output: words

SMT: *Strong Minimalist Thesis*

- The **SMT Parser** eschews all parsing algorithms and mechanisms, e.g. CKY, LR, etc.

That's the simplest imaginable condition of computational efficiency. If that makes expressions harder to process and even makes some thoughts impossible to express without circumlocution, too bad. Nature doesn't care.¹⁴

structure-dependence, relies on elementary properties of computational efficiency: adopting the simplest computational operation and the No-Tampering Condition (NTC), which basically says “don't add bells and whistles but keep to pure elegance.” That's the simplest imaginable condition of computational efficiency. If that makes expressions harder to process and even makes some thoughts impossible to express without circumlocution, too bad. Nature doesn't care.¹⁴

SOME ADVERTISING: theory & software

- Graduate seminar on *Syntax and Computation* (Fall semester 2026)
 - course materials will be freely available
 - a (*simplified*) SMT Parser will be freely available
 - (A few have it for testing/debugging)
 - *Currently, a bit uneven:*
 - *some simple parts are quite incomplete, others are quite sophisticated*
 - *many more examples to be put online*
 - *adding more theory*
 - *improving the computational model*
 - *work on a theory of EXT (language variation)*

LING 696A

Syntax and Computation

Administrivia

Instructor: Sandiway Fong Location: [redacted]

Although advanced, this course design has two tracks:

- **Syntax:** The *Strong Minimalist* developments based on Chomsky
- **Computation:** The *SMT Parser* will be made available.

I'll provide a gentle introduction to the play with the SMT Parser. It's an impl

```
swi-prolog -s code_spanish.prolog
% Janus 1.5.2 embeds Python 3.13.13 (main, Apr 8 2026, 18:23:14) [Clang 21]
Manual & tutorial: https://sandiway.arizona.edu Sandiway Fong (2024-2026)
Run: ?- parse(List,S0,L). (; for more answers)
List = [...words] (input), S0,L = syntactic object, [...] spellout (output)
See derivation: ?- report. ?- graph. (<- after parse/3) ?- summary. (short)
for a long report ?- retract(no_fill).
LEX expansion: (a) WordNet: ?- useMorphy. (noMorphy)
assert/retract/?- : use_morphy(C), for C={v,n,a,r}. (b) edit lex.prolog
(c) ?- addNoun(Word) (addTr/addUrg/addUnacc/addAdj/addPrep).
Welcome to SWI-Prolog (threaded, 64 bits, version 10.0.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

% Interactive session; added '.' to Python 'sys.path'
l01 ?- parse([él,le,gusta,a,ella],S,L).

Words: él le gusta a ella
Initial WS: ella_a appl_0 gustar_appl:0 v_gustar:pres INFL_v:3sg le é
INT/EXT: {C, {INFL_v:3sg, {v_gustar_appl:0 v_gustar:pres INFL_v:3sg le é
EXT: {[a,ella],y,él} 3sg-f pres gustar cl_le
Spellout: {[a,ella],y,él} le gusta
EXT: 3sg-f pres gustar {[a,ella],y,él} cl_le
Spellout: le gusta {[a,ella],y,él} cl_le
INT/EXT: inconsistent with input!
EXT: {C, {INFL_v:3sg, {v_gustar_appl:0 v_gustar:pres INFL_v:3sg le é
EXT: él 3sg-m pres gustar cl_le a ella
Spellout: él le gusta a ella
```

41

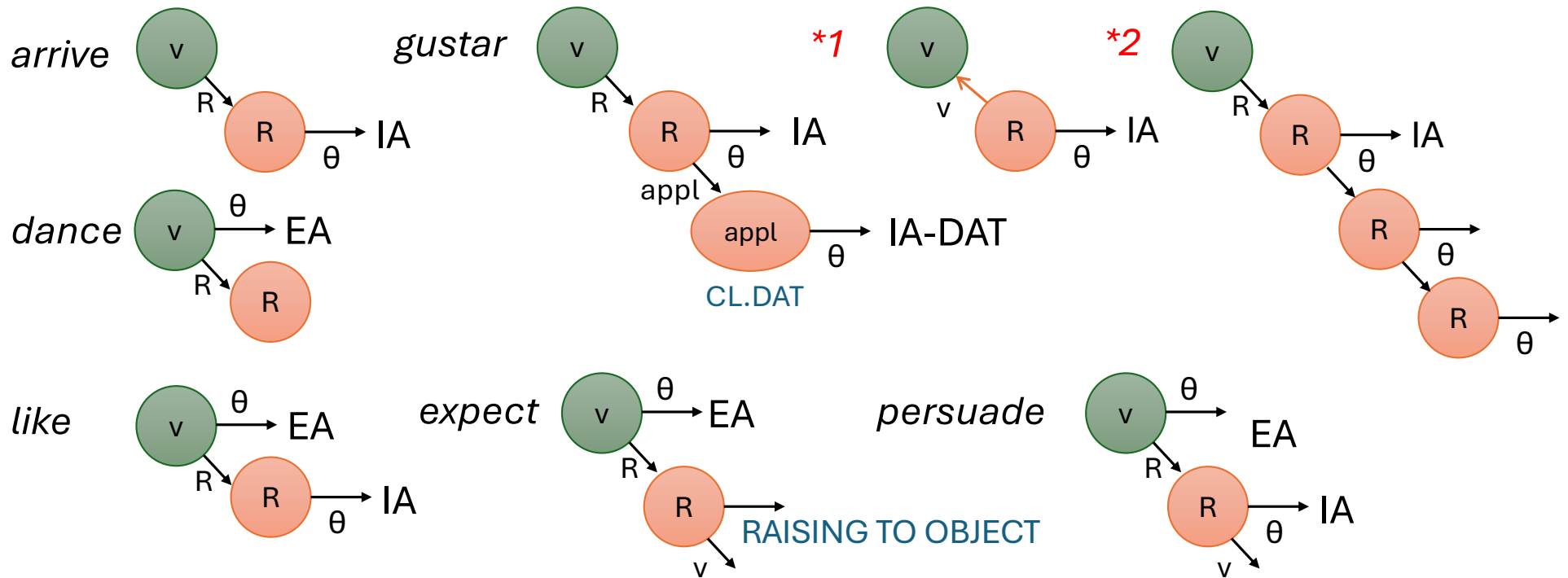
(Incomplete) Acknowledgements

- **(A-Z):** *Hilton Alers-Valentin* (NSF collaborator, U. of Puerto Rico at Mayaguez), *Noam Chomsky* (UA) *Jason Ginsburg* (Kyoto U.), *Carlos Krapp* (UA), Masumi Matsumoto (Osaka Kyoiku U.), *Masayuki Oishi* (Tohoku Gakuin U.), *Hiroshi Terada* (Osaka Kyoiku U.)
- **Spanish LEX:** *Alex Tubens* (NSF-funded Ph.D student), *Gabriela Quinones* (HLT Master's student)



Appendix

θ -configurations: *standard molecules*



θ -configurations: *standard molecules*

