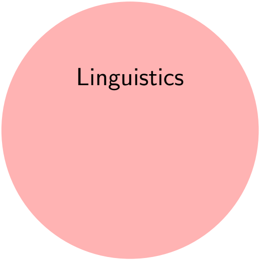
MG Parsing as a Window into Human Sentence Processing

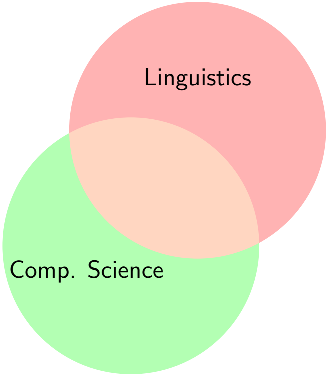
Aniello De Santo

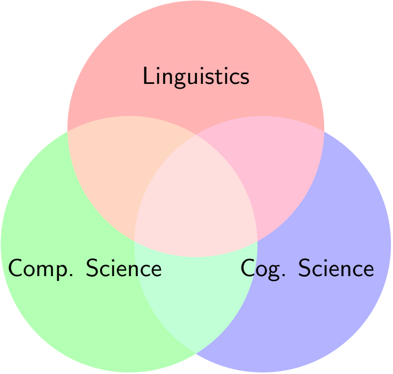
aniellodesanto.github.io
aniello.desanto@stonybrook.edu

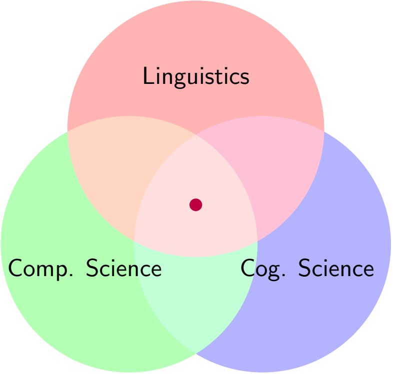
Get the slides!

UCLA
Feb 14, 2020

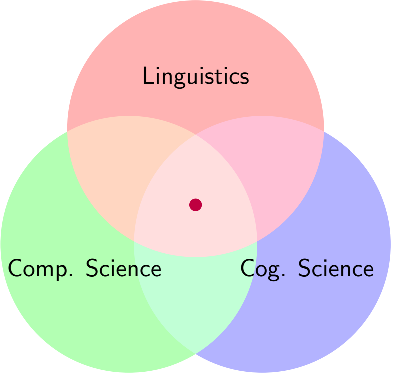

Linguistics


Linguistics
Comp. Science


Linguistics
Comp. Science
Cog. Science

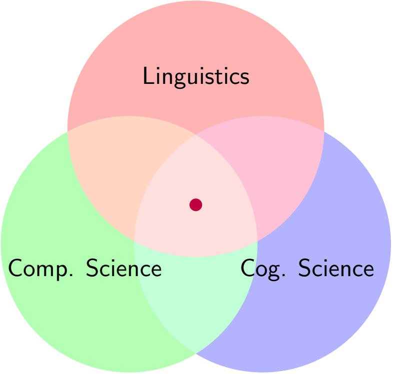


- Modeling processing difficulty (De Santo 2019; De Santo in prep.)
- Contrasting syntactic analyses (De Santo & Shafiei 2019)
- Gradience in acceptability judgments (De Santo 2020)
- Memory & generalized quantifiers (De Santo & Drury 2019, a.o)
- Subregularity of syntactic constraints (Graf & De Santo 2020)
- Subregular parallels across linguistic modules (Aksënova & De Santo 2017; De Santo & Graf 2019)
- Learnability (McMullin, Aksënova, De Santo 2018; De Santo 2018)
- Animal cognition (De Santo & Rawski 2020)



Processing

- Modeling processing difficulty (De Santo 2019; De Santo in prep.)
- Contrasting syntactic analyses (De Santo & Shafiei 2019)
- Gradience in acceptability judgments (De Santo 2020)
- Memory & generalized quantifiers (De Santo & Drury 2019, a.o)
- Subregularity of syntactic constraints (Graf & De Santo 2020)
- Subregular parallels across linguistic modules (Aksënova & De Santo 2017; De Santo & Graf 2019)
- Learnability (McMullin, Aksënova, De Santo 2018; De Santo 2018)
- Animal cognition (De Santo & Rawski 2020)

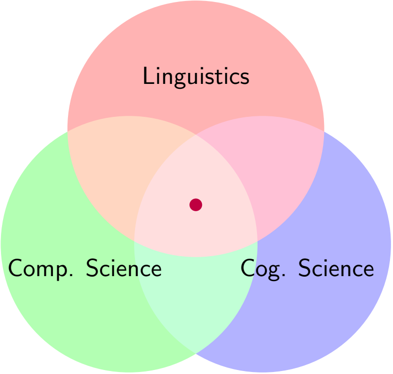


Processing

- Modeling processing difficulty (De Santo 2019; De Santo in prep.)
- Contrasting syntactic analyses (De Santo & Shafiei 2019)
- Gradience in acceptability judgments (De Santo 2020)
- Memory & generalized quantifiers (De Santo & Drury 2019, a.o)

FLT

- Subregularity of syntactic constraints (Graf & De Santo 2020)
- Subregular parallels across linguistic modules (Aksënova & De Santo 2017; De Santo & Graf 2019)
- Learnability (McMullin, Aksënova, De Santo 2018; De Santo 2018)
- Animal cognition (De Santo & Rawski 2020)



Processing / FLT

- Modeling processing difficulty (De Santo 2019; De Santo in prep.)
- Contrasting syntactic analyses (De Santo & Shafiei 2019)
- Gradience in acceptability judgments (De Santo 2020)
- Memory & generalized quantifiers (De Santo & Drury 2019, a.o)
- Subregularity of syntactic constraints (Graf & De Santo 2020)
- Subregular parallels across linguistic modules (Aksënova & De Santo 2017; De Santo & Graf 2019)
- Learnability (McMullin, Aksënova, De Santo 2018; De Santo 2018)
- Animal cognition (De Santo & Rawski 2020)

Let's Start with Data!

Asymmetries in Italian Relative Clauses

Italian speakers conform to the general cross-linguistic preference for SRC over ORC (Adani et al. 2010; Arosio et al. 2018)

(1)	Il	cavallo	che	ha	inseguito	i	leoni	
	The	horse	that	has	chased	the	lions	

"The horse that chased the lions" **SRC**

(2)	Il	cavallo	che	i	leoni	hanno	inseguito	
	The	horse	that	the	lions	have	chased	

"The horse that the lions chased" **ORC**

SRC > ORC

Postverbal Subjects and Ambiguity

Italian allows for postverbal subjects, making some sentences ambiguous (De Vincenzi 1991):

(3) Il cavallo che ha inseguito il leone
The horse that has chased the lion

a. "The horse that chased the lion" **SRC**

b. "The horse that the lion chased" **ORCp**

SRC > ORCp

Postverbal Subjects and Ambiguity

Italian allows for postverbal subjects, making some sentences ambiguous (De Vincenzi 1991):

(3) Il cavallo che ha inseguito il leone
The horse that has chased the lion

a. "The horse that chased the lion" **SRC**

b. "The horse that the lion chased" **ORCp**

SRC > ORCp

Postverbal Subjects and Ambiguity

Italian allows for postverbal subjects, making some sentences ambiguous (De Vincenzi 1991):

(3) Il cavallo che ha inseguito il leone
The horse that has chased the lion

a. "The horse that chased the lion" **SRC**

b. "The horse that the lion chased" **ORCp**

SRC > ORCp

Agreement can disambiguate:

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions

"The horse that the lions chased" **ORCp**

Asymmetries in Italian Relative Clauses

(1) Il cavallo che ha inseguito i leoni
The horse that has chased the lions
"The horse that chased the lions" **SRC**

(2) Il cavallo che i leoni hanno inseguito
The horse that the lions have chased
"The horse that the lions chased" **ORC**

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions
"The horse that the lions chased" **ORCp**

Processing Asymmetry (De Vincenzi 1991, Arosio et al. 2018, a.o.)

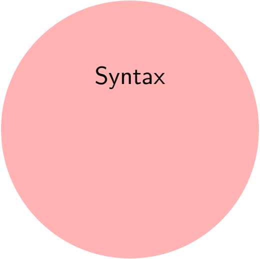
SRC > ORC > ORCp

One Big Question

(How much) does grammatical structure matter in sentence processing?

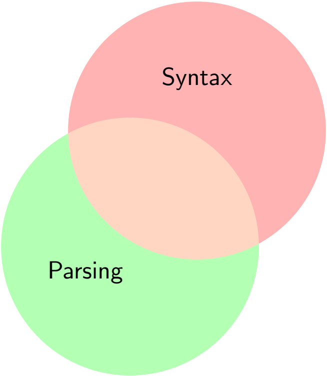
One Big Question

(How much) does grammatical structure matter in sentence processing?



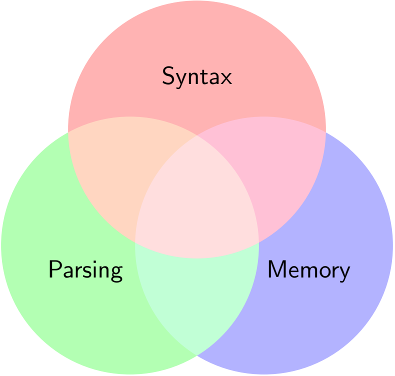
One Big Question

(How much) does grammatical structure matter in sentence processing?



One Big Question

(How much) does grammatical structure matter in sentence processing?



Forward to the Past

- What is the relation between grammatical operations and cognitive processes?

Derivational Theory of Complexity (Miller and Chomsky, 1963)

- Processing complexity ~ length of a derivation
  (Fodor & Garrett 1967; Berwick & Weinberg 1983)
- Essentially: there is a **cost** to mental computations.

- What is the right notion of syntactic derivation?
- What is costly? And why?

Forward to the Past

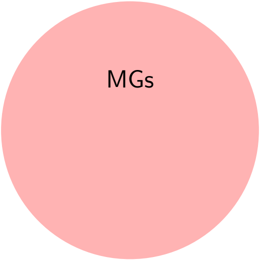
- What is the relation between grammatical operations and cognitive processes?

Derivational Theory of Complexity (Miller and Chomsky, 1963)

- Processing complexity $\sim$ length of a derivation (Fodor & Garrett 1967; Berwick & Weinberg 1983)
- Essentially: there is a **cost** to mental computations.

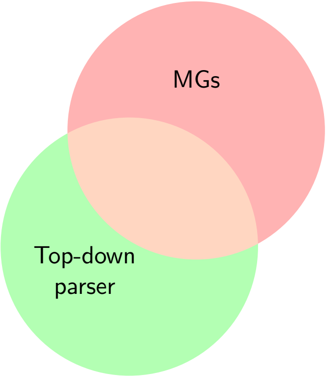
- What is the right notion of syntactic derivation?
- What is costly? And why?

A Formal Model of Sentence Processing



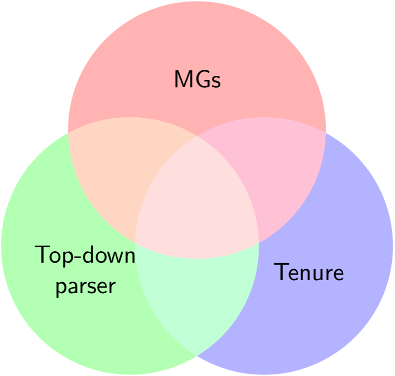
1. An explicit syntactic theory → Minimalist grammars (MGs)

A Formal Model of Sentence Processing



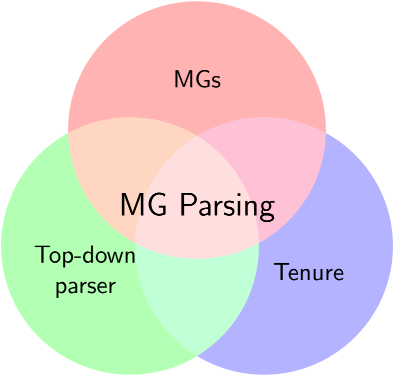
1. An explicit syntactic theory → Minimalist grammars (MGs)
2. A theory of how structures are built → top-down parser

A Formal Model of Sentence Processing



1. An explicit syntactic theory → Minimalist grammars (MGs)
2. A theory of how structures are built → top-down parser
3. A psychologically grounded linking theory → tenure

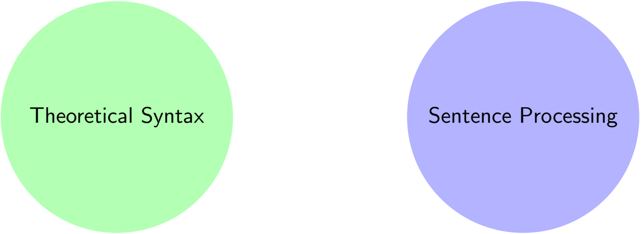
A Formal Model of Sentence Processing



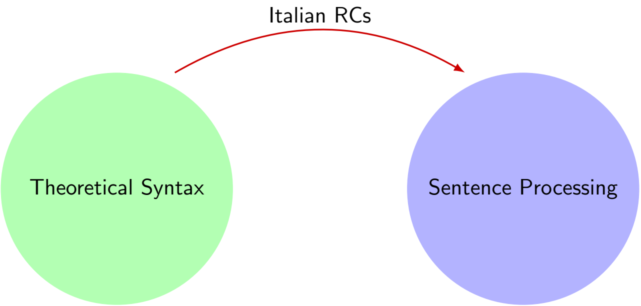
1. An explicit syntactic theory → Minimalist grammars (MGs)
2. A theory of how structures are built → top-down parser
3. A psychologically grounded linking theory → tenure

If you want to understand it, you can understand it!

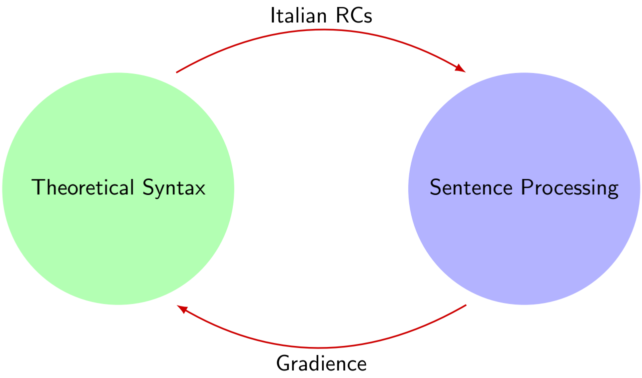
Building Bridges



Building Bridges



Building Bridges



Outline

Minimalist Grammars (MGs)

We need an explicit model of syntactic structures...

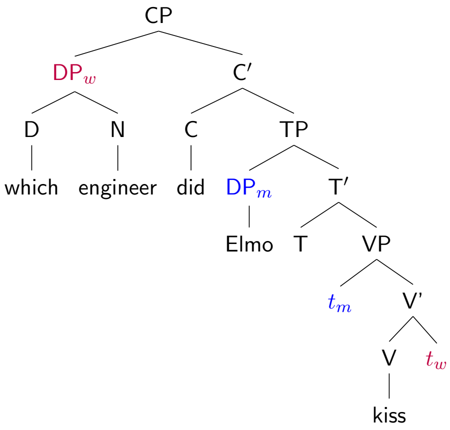
Ed Stabler

- Minimalist grammars (**MGs**): a formalization of Chomskyan syntax (Chomsky 1995; Stabler 1997)

Technical details!

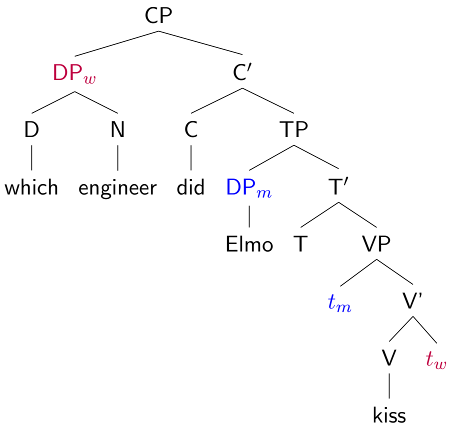
- Weakly equivalent to MCFGs
- Essentially: CFGs with a more complicated mapping from trees to strings
- REG tree language!

MG Syntax: Derivation Trees

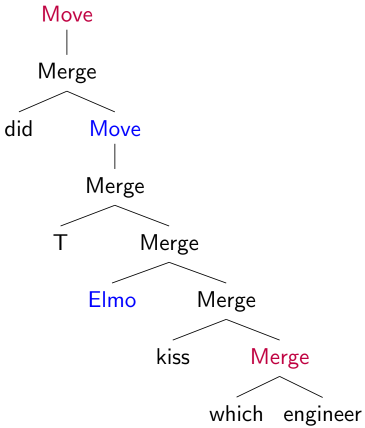


Phrase Structure Tree

MG Syntax: Derivation Trees

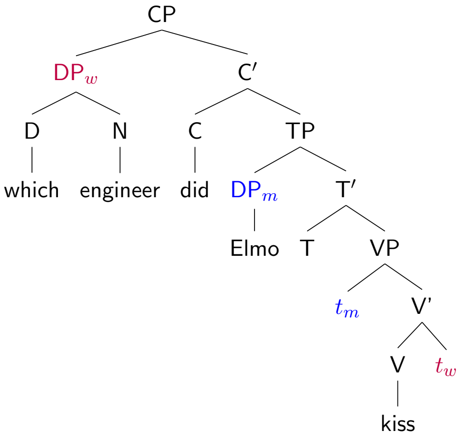


Phrase Structure Tree

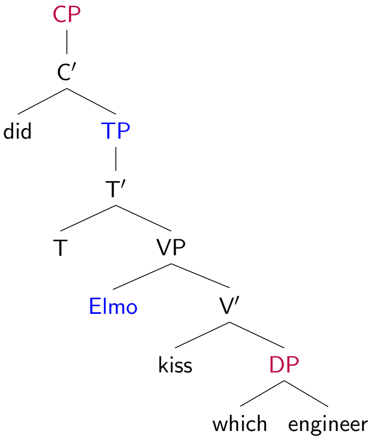


Derivation Tree

MG Syntax: Derivation Trees



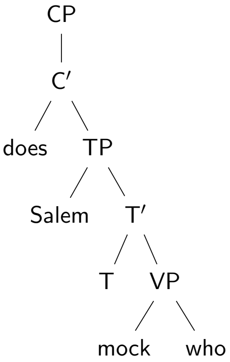
Phrase Structure Tree



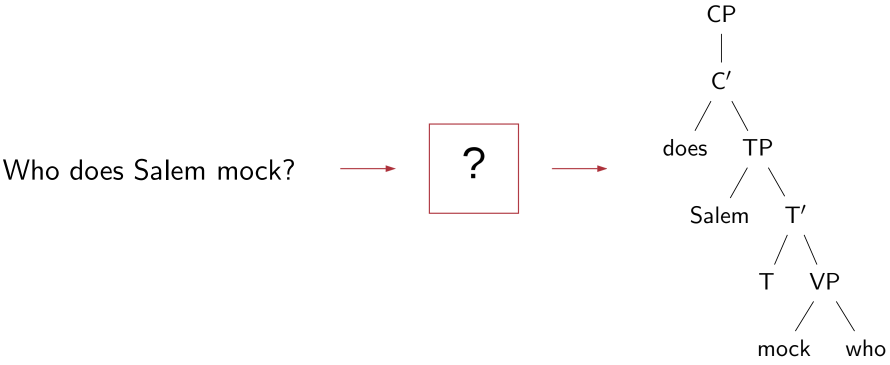
Derivation Tree

The Job of a Parser

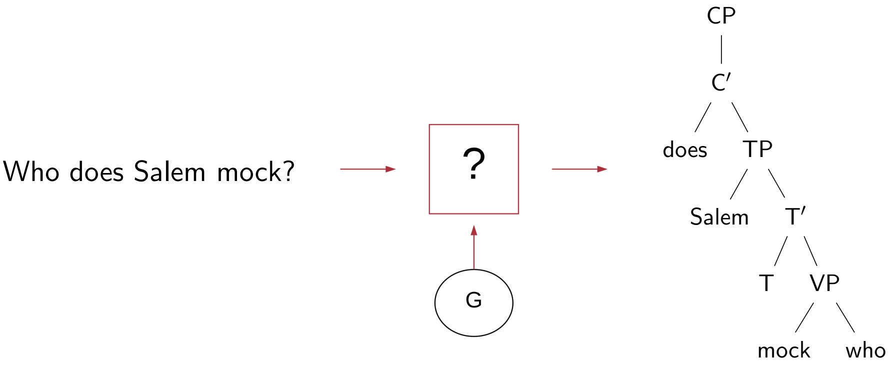
Who does Salem mock?



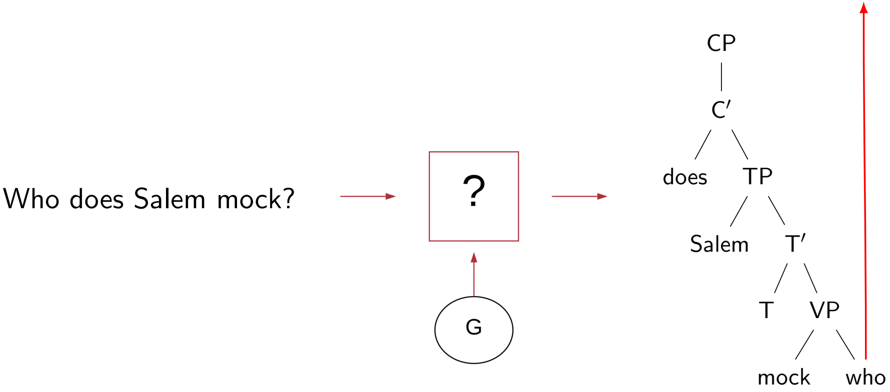
The Job of a Parser



The Job of a Parser



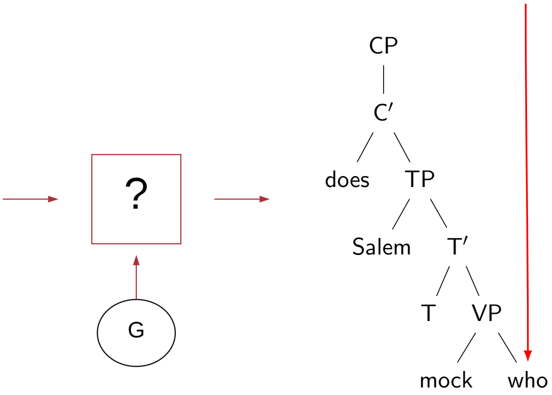
The Job of a Parser



- Bottom-up

The Job of a Parser

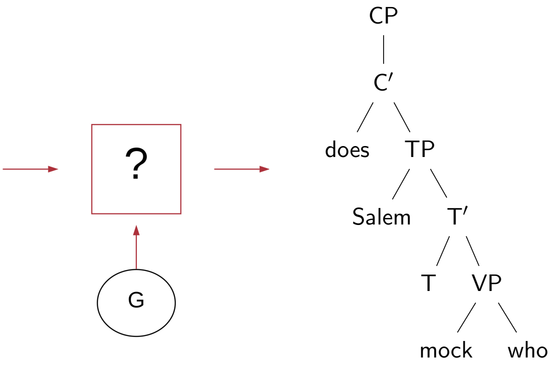
Who does Salem mock?



- Bottom-up
- Top-down

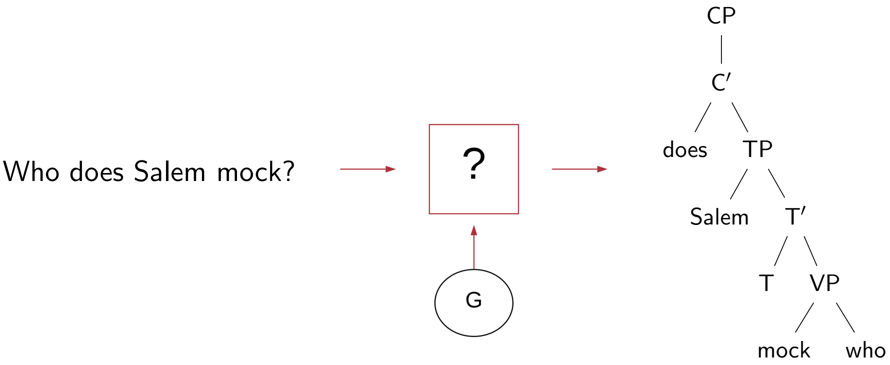
The Job of a Parser

Who does Salem mock?



- Bottom-up
- Top-down
  - Psychologically plausible(-ish)

The Job of a Parser



- Bottom-up
- Top-down
  - Psychologically plausible(-ish)
  - Insight: We can build lexicalized grammars top-down!
  - Assumption: Parser as an oracle!

Top-Down Parsing: The Intuition

Who does Salem mock?

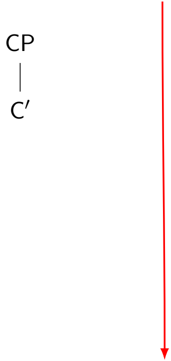
Top-Down Parsing: The Intuition

CP

Who does Salem mock?

- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure ≈ how much memory is used!

Top-Down Parsing: The Intuition

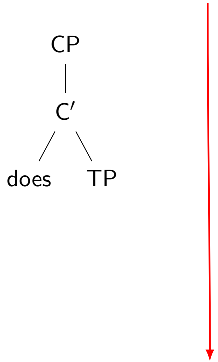


Who does Salem mock?

- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure ≈ how much memory is used!

Top-Down Parsing: The Intuition

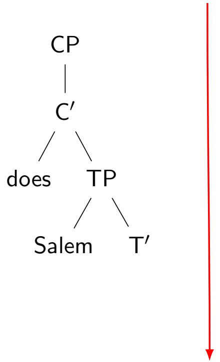
Who does Salem mock?



- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure ≈ how much memory is used!

Top-Down Parsing: The Intuition

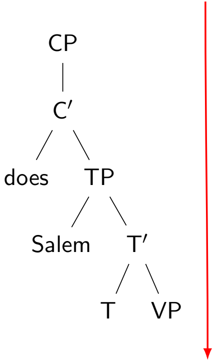
Who does Salem mock?



- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure ≈ how much memory is used!

Top-Down Parsing: The Intuition

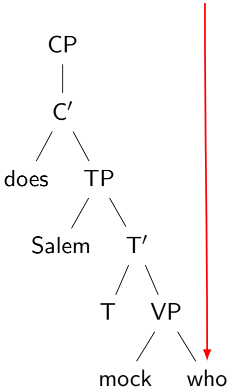
Who does Salem mock?



- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure $\approx$ how much memory is used!

Top-Down Parsing: The Intuition

Who does Salem mock?



- Builds the structure from top to bottom
- Takes elements in an out of memory
- Complexity of the structure ≈ how much memory is used!

Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 CP is conjectured
step 2 CP expands to C′
step 3 C′ expands to does and TP
step 4 TP expands to Salem and T′
step 5 T′ expands to T and VP
step 6 VP expands to mock and who
step 7 who is found
step 8 does is found
step 9 Salem is found
step 10 T is found
step 11 mock is found

Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

^{1}CP

- • Who • does • Salem • T • mock

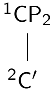
step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found

Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock



step 1 *CP* is conjectured
step 2 *CP* expands to *C′*
step 3 *C′* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T′*
step 5 *T′* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found

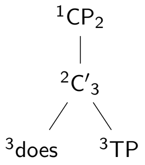
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



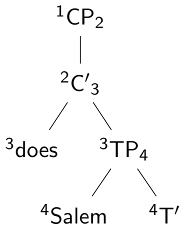
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1	*CP* is conjectured
step 2	*CP* expands to *C'*
step 3	*C'* expands to *does* and *TP*
step 4	*TP* expands to *Salem* and *T'*
step 5	*T'* expands to *T* and *VP*
step 6	*VP* expands to *mock* and *who*
step 7	*who* is found
step 8	*does* is found
step 9	*Salem* is found
step 10	*T* is found
step 11	*mock* is found



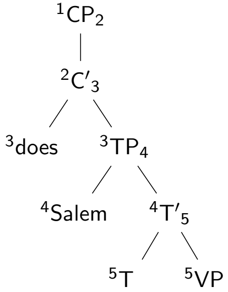
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



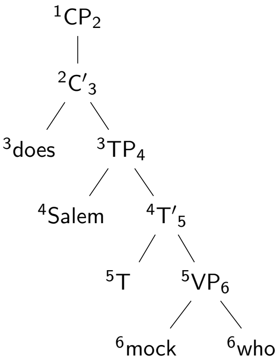
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



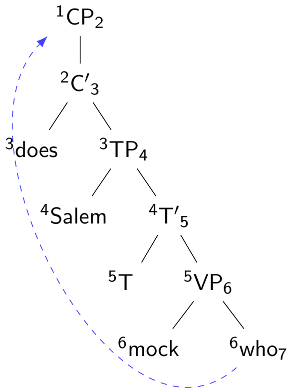
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1	*CP* is conjectured
step 2	*CP* expands to *C'*
step 3	*C'* expands to *does* and *TP*
step 4	*TP* expands to *Salem* and *T'*
step 5	*T'* expands to *T* and *VP*
step 6	*VP* expands to *mock* and *who*
step 7	*who* is found
step 8	*does* is found
step 9	*Salem* is found
step 10	*T* is found
step 11	*mock* is found



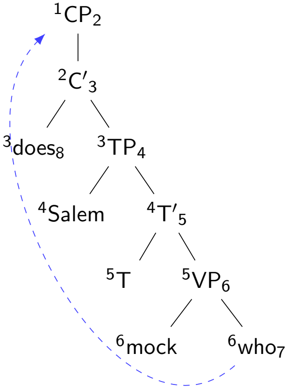
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



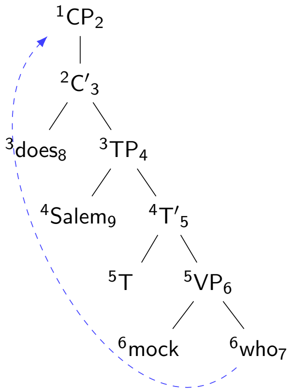
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



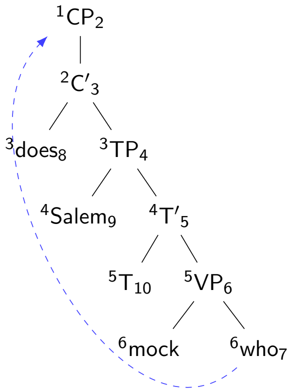
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1 *CP* is conjectured
step 2 *CP* expands to *C'*
step 3 *C'* expands to *does* and *TP*
step 4 *TP* expands to *Salem* and *T'*
step 5 *T'* expands to *T* and *VP*
step 6 *VP* expands to *mock* and *who*
step 7 *who* is found
step 8 *does* is found
step 9 *Salem* is found
step 10 *T* is found
step 11 *mock* is found



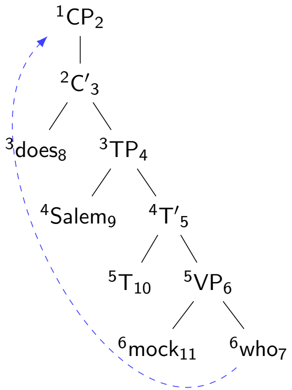
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1	*CP* is conjectured
step 2	*CP* expands to *C'*
step 3	*C'* expands to *does* and *TP*
step 4	*TP* expands to *Salem* and *T'*
step 5	*T'* expands to *T* and *VP*
step 6	*VP* expands to *mock* and *who*
step 7	*who* is found
step 8	*does* is found
step 9	*Salem* is found
step 10	*T* is found
step 11	*mock* is found



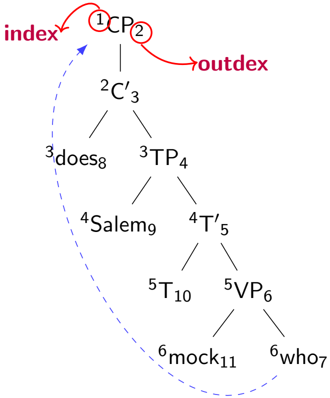
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1	*CP* is conjectured
step 2	*CP* expands to *C'*
step 3	*C'* expands to *does* and *TP*
step 4	*TP* expands to *Salem* and *T'*
step 5	*T'* expands to *T* and *VP*
step 6	*VP* expands to *mock* and *who*
step 7	*who* is found
step 8	*does* is found
step 9	*Salem* is found
step 10	*T* is found
step 11	*mock* is found



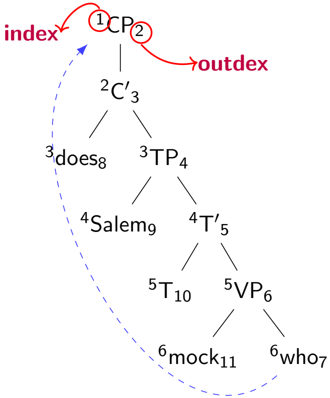
Incremental Top-Down Parsing

Technical details!

- String-driven recursive descent parser (Stabler 2013)

- • Who • does • Salem • T • mock

step 1	*CP* is conjectured
step 2	*CP* expands to *C'*
step 3	*C'* expands to *does* and *TP*
step 4	*TP* expands to *Salem* and *T'*
step 5	*T'* expands to *T* and *VP*
step 6	*VP* expands to *mock* and *who*
step 7	*who* is found
step 8	*does* is found
step 9	*Salem* is found
step 10	*T* is found
step 11	*mock* is found



Index and Outdex are our connection to memory!

Memory-Based Complexity Metrics

- **Memory usage**:
  (Kobele et al. 2012; Gibson, 1998)

Tenure: How long a node is kept in memory
Size: How much information is stored in a node
$\Rightarrow$ Intuitively, the length of its movement dependency!

- Formalized into **complexity metrics**

MaxTenure: $max(\{\text{tenure-of}(n) | n \text{ a node of the tree}\})$
SumSize: $\sum_{m \in M} size(m)$
Ranked: $\langle MaxTenure, SumSize \rangle$

Greg Kobele

Sabrina Gerth

John Hale

Memory-Based Complexity Metrics

- **Memory usage**:
  (Kobele et al. 2012; Gibson, 1998)

Tenure How long a node is kept in memory
Size How much information is stored in a node
$\Rightarrow$ Intuitively, the length of its movement dependency!

- Formalized into **complexity metrics**

MaxTenure $max(\{\text{tenure-of}(n) | n \text{ a node of the tree}\})$

SumSize $\sum_{m \in M} size(m)$

Ranked $\langle MaxTenure, SumSize \rangle$

Greg Kobele

Sabrina Gerth

John Hale

Memory-Based Complexity Metrics

- **Memory usage**:
  (Kobele et al. 2012; Gibson, 1998)

Tenure	How long a node is kept in memory
Size	How much information is stored in a node ⇒ Intuitively, the length of its movement dependency!

- Formalized into **complexity metrics**

MaxTenure	$max(\{\text{tenure-of}(n) \mid n \text{ a node of the tree}\})$
SumSize	$\sum_{m \in M} size(m)$
Ranked	$\langle MaxTenure, SumSize \rangle$

Greg Kobele

Sabrina Gerth

John Hale

Processing Asymmetries All the Way Down

<MaxT,SumS> makes correct predictions cross-linguistically!

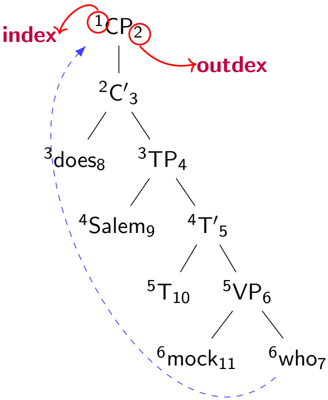
Across Many Constructions

- Right > center embedding (Kobele et al. 2012)
- Crossing > nested dependencies (Kobele et al. 2012)
- SC-RC > RC-SC (Graf & Marcinek 2014)
- SRC > ORC (Graf et al. 2017)
- Postverbal subjects in Italian (De Santo 2019)
- Persian attachment ambiguities (De Santo & Shafiei 2019)
- Gradient acceptability (De Santo 2020)

Across Languages

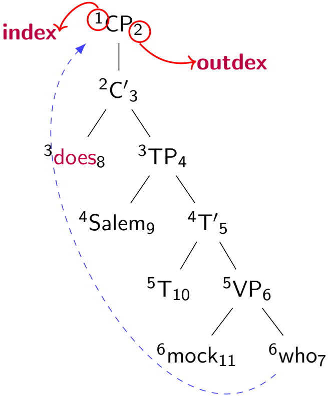
- English, German, Italian
- Korean, Japanese
- Mandarin Chinese
- Persian

Computing Metrics: An Example



Tenure how long a node is kept in memory

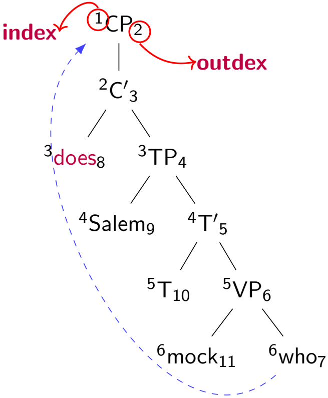
Computing Metrics: An Example



Tenure how long a node is kept in memory

Tenure($does$**)** $= 8 - 3 = 5$

Computing Metrics: An Example



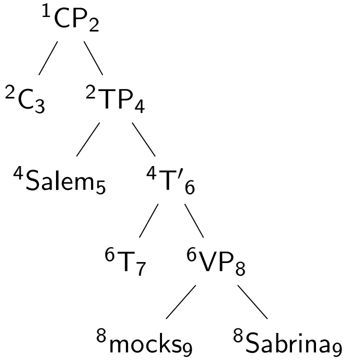
Tenure how long a node is kept in memory

Tenure($does$**)** $= 8 - 3 = 5$

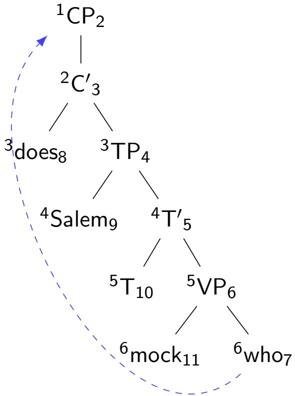
MaxTenure $= max\{$**Tenure**$(does)$, **Tenure**$(Salem), \dots\} = 5$

Contrasting Derivations

MaxTenure $= 2$



MaxTenure $= 5$



Automatizing Helps!

mgproc: A Python Package for MG Processing Research

This is a collection of Python3 scripts to facilitate the investigation of human processing from the perspective of Minimalist grammars (MGs).

Background

MGs were developed in Stabler (1997) as a formalization of Chomsky's Minimalist program. A top-down parser for MGs is defined in Stabler (2013) and has been implemented in a number of languages. A number of subsequent works have successfully used this parser to make predictions about relative difficulty in sentence processing. Good starting points with a review of the previous literature are Gerth (2015) and Graf et al. (to appear).

- Gerth, Sabrina: Memory Limitations in Sentence Comprehension
- Graf, Thomas, James Monette, and Chong Zhang (to appear): Relative Clauses as a Benchmark for Minimalist Parsing (link to be added soon)
- Stabler, Edward (1997): Derivational Minimalism
- Stabler, Edward (2013): Two Models of Minimalist, Incremental Syntactic Analysis

Quick Start Guide

With *mgproc* you can easily compare MG derivation trees with respect to thousands of complexity m processing. The scripts integrate well with a LaTeX-centric workflow, following the ideal of OpenScie publication form a cohesive unit. Usually a parsed derivation tree is specified by four files. Assuming `foo`, we have:

- Open source ⇒ in prep. for *Journal of Open Source Software*
- User-friendly!
- Easy to modify!

Summary of the Approach

General Idea

(Kobele et al. 2012; Gerth 2015; Graf et al. 2017)

1. Pick two competing derivations
2. Evaluate metrics over each
   - Lowest score means easiest!
3. Compare parser's prediction to experimental data

Remember!

If you want to understand it, you can understand it!

Reminder: Asymmetries in Italian Relative Clauses

(1) Il cavallo che ha inseguito i leoni
The horse that has chased the lions
"The horse that chased the lions" **SRC**

(2) Il cavallo che i leoni hanno inseguito
The horse that the lions have chased
"The horse that the lions chased" **ORC**

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions
"The horse that the lions chased" **ORCp**

Processing Asymmetry (De Vincenzi 1991, Arosio et al. 2018, a.o.)

SRC > ORC > ORCp

Modeling Assumptions

Reminder:

- Parsing strategy
  ⇒ Top-down parser
- Complexity Metrics
  ⇒ MaxTenure and SumSize

Degrees of freedom: Syntactic analyses

1. RC constructions → (Kayne 1994)
2. Postverbal subjects → (Belletti & Leonini 2004)

Modeling Assumptions

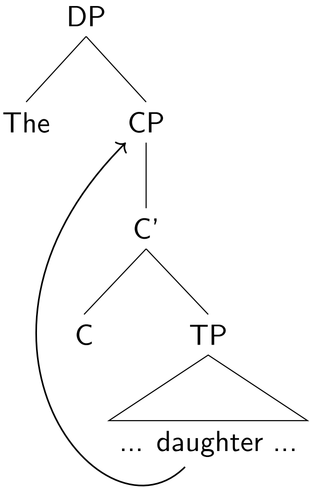
Reminder:

- Parsing strategy
  ⇒ Top-down parser
- Complexity Metrics
  ⇒ MaxTenure and SumSize

Degrees of freedom: Syntactic analyses

1. RC constructions → (Kayne 1994)
2. Postverbal subjects → (Belletti & Leonini 2004)

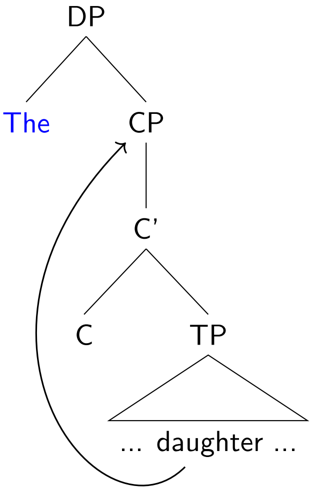
Kayne's Promotion Analysis (Kayne 1994)



- RC is selected by an external D^0
- the RC head is a nominal constituent
- the RC head raises from its base position to [Spec, CP]

$[_{DP}$ The $[_{CP}$ daughter$_i$ [that t_i was on the balcony]]]

Kayne's Promotion Analysis (Kayne 1994)

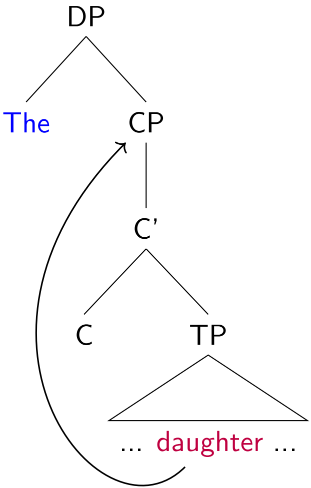


- RC is selected by an external D^0
- the RC head is a nominal constituent
- the RC head raises from its base position to [Spec, CP]

[$_{DP}$ The [$_{CP}$ daughter$_i$ [that t_i was on the balcony]]]

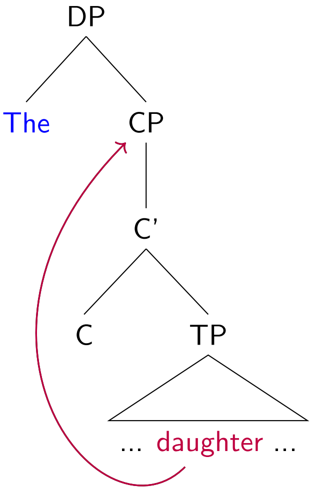
Kayne's Promotion Analysis (Kayne 1994)

- RC is selected by an external D^0
- the RC head is a nominal constituent
- the RC head raises from its base position to [Spec, CP]



[$_{DP}$ The [$_{CP}$ daughter$_i$ [that t_i was on the balcony]]]

Kayne's Promotion Analysis (Kayne 1994)



- RC is selected by an external D^0
- the RC head is a nominal constituent
- the RC head raises from its base position to [Spec, CP]

[$_{DP}$ The [$_{CP}$ daughter$_i$ [that t_i was on the balcony]]]

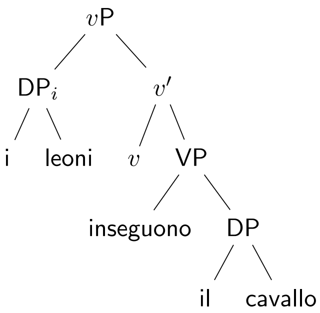
Postverbal Subjects (Belletti & Leonini 2004)

(5) Inseguono il cavallo i leoni
Chase the horse the lions
"The lions chase the horse"

- the subject DP raises to Spec, FocP
- The whole vP raises to Spec,TopP

Technical details!

- an expletive *pro* is base generated in Spec,TP



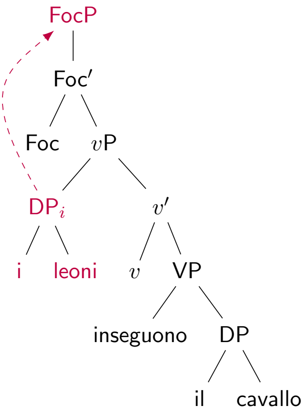
Postverbal Subjects (Belletti & Leonini 2004)

(6) Inseguono il cavallo i leoni
Chase the horse the lions
"The lions chase the horse"

- the subject DP raises to Spec, FocP
- The whole vP raises to Spec,TopP

Technical details!

- an expletive *pro* is base generated in Spec,TP



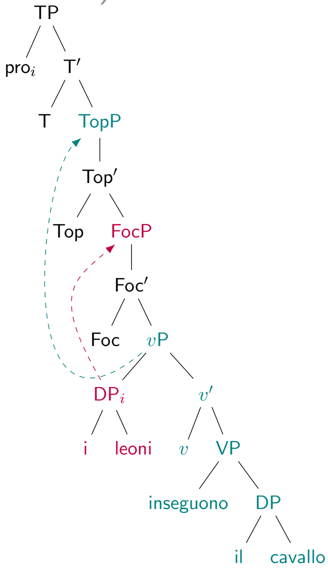
Postverbal Subjects (Belletti & Leonini 2004)

(7) Inseguono il cavallo i leoni
Chase the horse the lions
"The lions chase the horse"

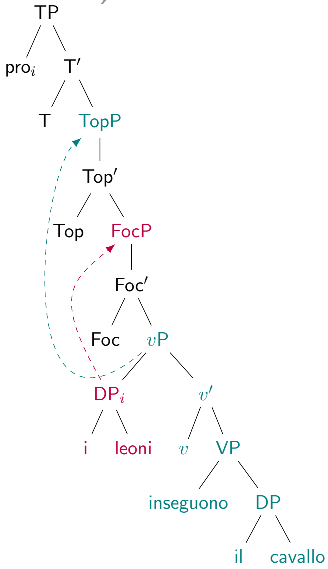
- the subject DP raises to Spec, FocP
- The whole vP raises to Spec,TopP

Technical details!

- an expletive *pro* is base generated in Spec,TP



Postverbal Subjects (Belletti & Leonini 2004)



(7) Inseguono il cavallo i leoni
Chase the horse the lions
"The lions chase the horse"

- the subject DP raises to Spec, FocP
- The whole vP raises to Spec,TopP

Technical details!

- an expletive *pro* is base generated in Spec,TP

Modeling Results

(1) Il cavallo che ha inseguito i leoni
The horse that has chased the lions
"The horse that chased the lions" **SRC**

(2) Il cavallo che i leoni hanno inseguito
The horse that the lions have chased
"The horse that the lions chased" **ORC**

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions
"The horse that the lions chased" **ORCp**

SRC $>$ ORC $>$ ORCp

Modeling Results

(1) Il cavallo che ha inseguito i leoni
The horse that has chased the lions
"The horse that chased the lions" **SRC**

(2) Il cavallo che i leoni hanno inseguito
The horse that the lions have chased
"The horse that the lions chased" **ORC**

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions
"The horse that the lions chased" **ORCp**

	SRC	>	ORC	>	ORCp
MaxTenure	8/che		11/ha		16/Foc
SumSize	18		24		31

Modeling Results

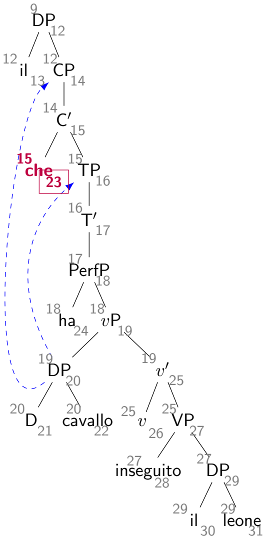
(1) Il cavallo che ha inseguito i leoni
The horse that has chased the lions
"The horse that chased the lions" **SRC**

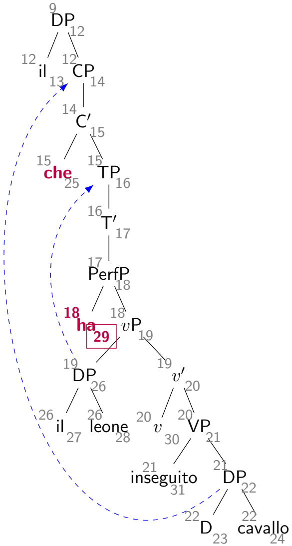
(2) Il cavallo che i leoni hanno inseguito
The horse that the lions have chased
"The horse that the lions chased" **ORC**

(4) Il cavallo che hanno inseguito i leoni
The horse that have chased the lions
"The horse that the lions chased" **ORCp**

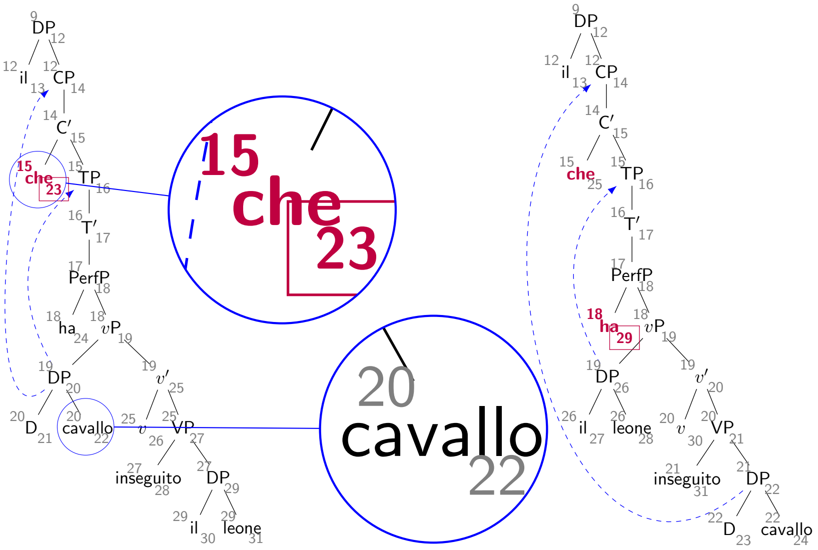
	SRC	>	ORC	>	ORCp	
MaxTenure	8/che		11/ha		16/Foc	✓
SumSize	18		24		31	✓

Results: SRC > ORC

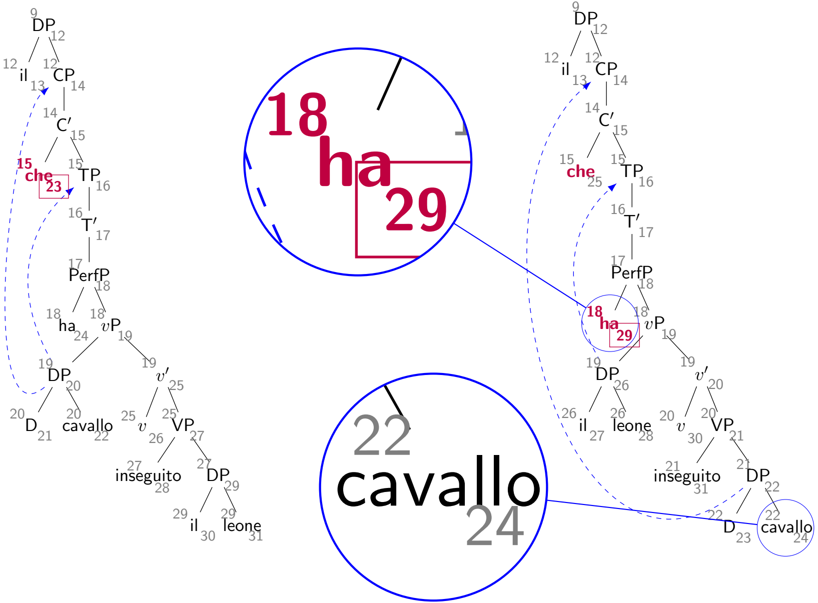




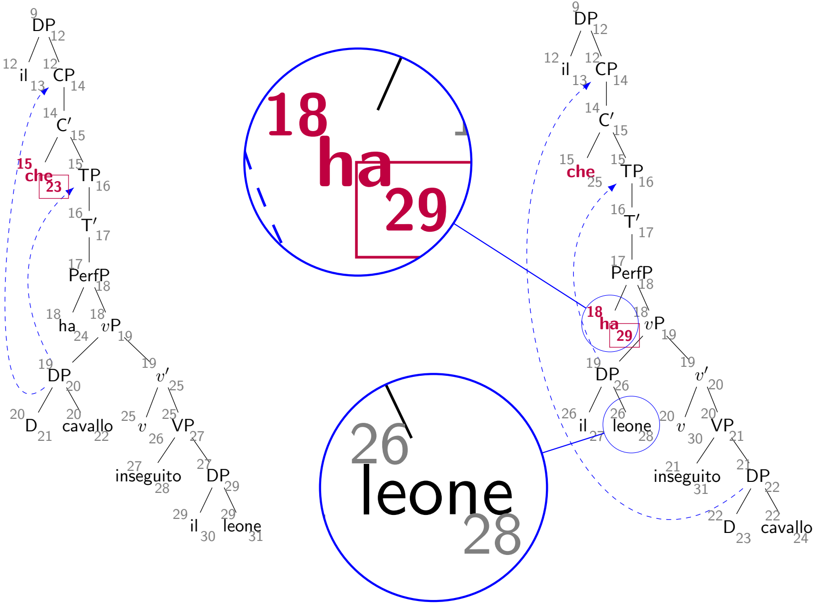
Results: SRC > ORC



Results: SRC > ORC



Results: SRC > ORC



Summary of Results (De Santo 2019)

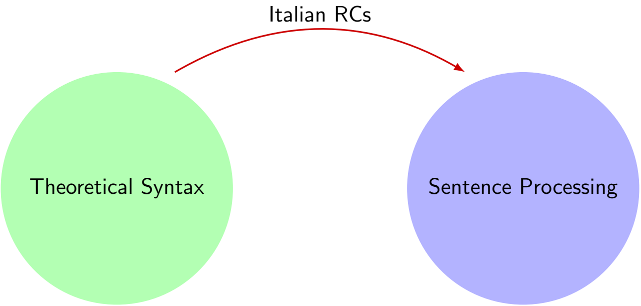
Clause Type	<MaxTenure,SumSize>
obj. SRC > ORC	✓
obj. SRC > ORCp	✓
obj. ORC > ORCp	✓
subj. SRC > ORC	✓
subj. SRC > ORCp	✓
subj. ORC > ORCp	✓
matrix SVO > VOS	✓
VS unacc > VS unerg	✓

Table: Predictions of the MG parser by contrast.

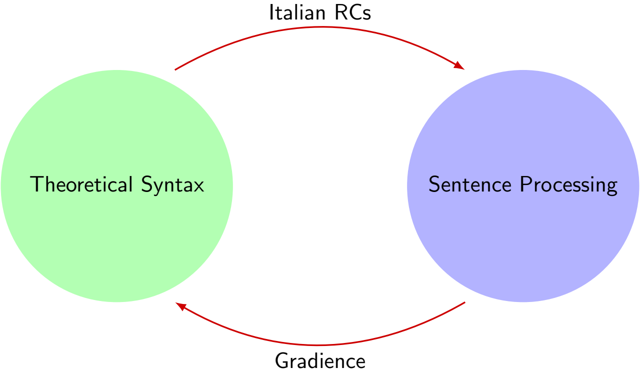
Interim Summary

- Asymmetries in Italian postverbal subject constructions
  - Derived just from **(fine-grained) structural differences**!
  - **Ongoing**: expand range of syntactic analyses;
  - **Ongoing**: cross-linguistic comparisons.
- <MaxT,SumS> gives consistent results!
  - Right vs. center embedding, attachment ambiguities, relative clause preferences
  - English, German, Korean, Japanese, Persian, Mandarin Chinese
  - More?

Moving on



Moving on


Italian RCs
Theoretical Syntax
Sentence Processing
Gradience

Acceptability and Grammaticality

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?

Acceptability and Grammaticality

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?

> *One way to test the adequacy of a grammar proposed for [language] L is to determine whether or not the sequences that it generates are actually grammatical, i.e., acceptable to a native speaker.*
>
> *(Chomsky 1957)*

Acceptability and Grammaticality

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?

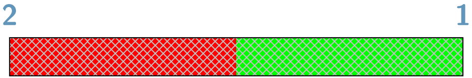
One way to test the adequacy of a grammar proposed for [language] L is to determine whether or not the sequences that it generates are actually grammatical, i.e., acceptable to a native speaker.

(Chomsky 1957)

Acceptability judgments ≈ Grammaticality judgments

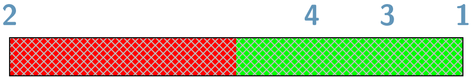
Gradience in Acceptability Judgments

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?



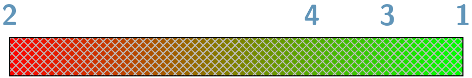
Gradience in Acceptability Judgments

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?
3. Who *t* thinks that John bought a car?
4. Who *t* wonders whether John bought a car?



Gradience in Acceptability Judgments

1. What do you think that John bought *t*?
2. *What do you wonder whether John bought *t*?
3. Who *t* thinks that John bought a car?
4. Who *t* wonders whether John bought a car?



Gradient Acceptability and Categorical Grammars

Acceptability judgments are not binary but *gradient*:

> *An adequate linguistic theory will have to recognize degrees of grammaticalness [...] there is little doubt that speakers can fairly consistently order new utterances, never previously heard, with respect to their degree of belongingness to the language.*
>
> *(Chomsky 1975: 131-132)*

But mainstream syntactic theories rely on categorical grammars!

Gradient Acceptability and Categorical Grammars

Acceptability judgments are not binary but *gradient*:

> *An adequate linguistic theory will have to recognize degrees of grammaticalness [...] there is little doubt that speakers can fairly consistently order new utterances, never previously heard, with respect to their degree of belongingness to the language.*
>
> (Chomsky 1975: 131-132)

But mainstream syntactic theories rely on categorical grammars!

Models of Gradience

(At least two) theories of gradience:

- Gradience incorporated in the grammar
  (Keller 2000; Featherston 2005; Lau et al. 2014)
- Gradience due to extra-grammatical factors
  (Chomsky 1975; Schütze 1996)

The contribution of formal models?

Quantify what each approach needs to account for the data:

- Additional syntactic assumptions
- Additional complexity in acquisition, processing strategies, etc.

Models of Gradience

(At least two) theories of gradience:

- Gradience incorporated in the grammar
  (Keller 2000; Featherston 2005; Lau et al. 2014)
- Gradience due to extra-grammatical factors
  (Chomsky 1975; Schütze 1996)

The contribution of formal models?

Quantify what each approach needs to account for the data:

- Additional syntactic assumptions
- Additional complexity in acquisition, processing strategies, etc.

(Quantitative) Models of Gradience

Gradient Grammars (Keller 2000; Lau et al. 2014)

- OT-style constraint ranking
- Probabilistic grammars

Extra-grammatical Factors (Chomsky 1975; Schütze 1996)

- Processing effects
  - Plausibility
  - Working memory limitations
  - **But:** few models for quantitative predictions!

Hypothesis

We can use the MG parser to test the relation between categorical grammar, processing difficulty, and gradience!

(Quantitative) Models of Gradience

Gradient Grammars (Keller 2000; Lau et al. 2014)

- OT-style constraint ranking
- Probabilistic grammars

Extra-grammatical Factors (Chomsky 1975; Schütze 1996)

- Processing effects
  - Plausibility
  - Working memory limitations
  - **But:** few models for quantitative predictions!

Hypothesis

We can use the MG parser to test the relation between categorical grammar, processing difficulty, and gradience!

Modeling Gradience with an MG Parser

The model is the same as before

1. A formal model of syntax → Minimalist grammars (MGs)
2. A theory of how structures are built → MG parser
3. A linking theory: higher memory cost ⇒ lower acceptability

- Sensitive to fine-grained structural differences!
- Minimal, pairwise comparisons are maximally interpretable!

A proof-of-concept:

- Variation of Island effects in English (Sprouse et al. 2012)

A Proof of Concept: Island Effects

1. What do you think that John bought *t*?
2. What do you wonder whether John bought *t*?
3. Who *t* thinks that John bought a car?
4. Who *t* wonders whether John bought a car?

Results in pairwise comparisons ideal for the MG parser

A Proof of Concept: Island Effects

1. What do you think that John bought *t*?
2. What do you wonder whether John bought *t*?
3. Who *t* thinks that John bought a car?
4. Who *t* wonders whether John bought a car?

Gradience in Islands: Sprouse et al. (2012)

A factorial design for islands effects:

1. GAP POSITION: Matrix vs. Embedded
2. STRUCTURE: Island vs. Non-Island

(Kluender & Kutas 1993)

Jon Sprouse

Results in pairwise comparisons ideal for the MG parser

A Proof of Concept: Island Effects

1. What do you think that John bought *t*? — Non-Island — Embedded
2. What do you wonder whether John bought *t*? — Island — Embedded
3. Who *t* thinks that John bought a car? — Non-Island — Matrix
4. Who *t* wonders whether John bought a car? — Island — Matrix

Gradience in Islands: Sprouse et al. (2012)

A factorial design for islands effects:

1. GAP POSITION: Matrix vs. Embedded
2. STRUCTURE: Island vs. Non-Island
   (Kluender & Kutas 1993)

Jon Sprouse

Results in pairwise comparisons ideal for the MG parser

A Proof of Concept: Island Effects

1. What do you think that John bought *t*? — Non-Island — Embedded
2. What do you wonder whether John bought *t*? — Island — Embedded
3. Who *t* thinks that John bought a car? — Non-Island — Matrix
4. Who *t* wonders whether John bought a car? — Island — Matrix

Gradience in Islands: Sprouse et al. (2012)

A factorial design for islands effects:

1. Gap Position: Matrix vs. Embedded
2. Structure: Island vs. Non-Island

(Kluender & Kutas 1993)

Jon Sprouse

Results in pairwise comparisons ideal for the MG parser

Sprouse at al. (2012)

Four island types

Subject islands

- What do you think the speech about *t* interrupted the show about global warming?

Adjunct islands

- What do you laugh if John leaves *t* at the office?

Complex NP islands

- What did you make the claim that John bought *t*?

Whether islands

- What do you wonder whether John bought *t*?

Gap Position × Structure

1. Matrix vs. Embedded
2. Island vs. Non-Island

Sprouse at al. (2012)

FOUR ISLAND TYPES

Subject islands

- What do you think the speech about *t* interrupted the show about global warming?

Adjunct islands

- What do you laugh if John leaves *t* at the office?

Complex NP islands

- What did you make the claim that John bought *t*?

Whether islands

- What do you wonder whether John bought *t*?

GAP POSITION × STRUCTURE

1. Matrix vs. Embedded
2. Island vs. Non-Island

Modeling Results (De Santo 2020)

Island Type	Sprouse et al. (2012)			MG Parser
Subj. Island 1	Subj. — Non Isl.	>	Obj. — Non Isl.	✓
	Subj. — Non Isl.	>	Obj. — Isl.	✓
	Subj. — Non Isl.	>	Subj. — Isl.	✓
	Obj. — Non Isl.	>	Obj. — Isl.	✓
	Obj. — Non Isl.	>	Subj. — Isl.	✓
	Obj. — Isl.	>	Subj. — Isl.	×
Subj. Island 2	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	>	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓
Adj. Island	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	>	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓
CNP Island	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	=	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓

Modeling Results (De Santo 2020)

Island Type	Sprouse et al. (2012)			MG Parser
Subj. Island 1	Subj. — Non Isl.	>	Obj. — Non Isl.	✓
	Subj. — Non Isl.	>	Obj. — Isl.	✓
	Subj. — Non Isl.	>	Subj. — Isl.	✓
	Obj. — Non Isl.	>	Obj. — Isl.	✓
	Obj. — Non Isl.	>	Subj. — Isl.	✓
	Obj. — Isl.	**>**	**Subj. — Isl.**	×
Subj. Island 2	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	>	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓
Adj. Island	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	>	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓
CNP Island	Matrix — Non Isl.	>	Emb. — Non Isl.	✓
	Matrix — Non Isl.	=	Matrix — Isl.	✓
	Matrix — Non Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Emb. — Isl.	✓
	Matrix — Isl.	>	Matrix — Isl.	✓
	Emb. — Non Isl.	>	Emb. — Isl.	✓

TL;DR

Success in all cases but one!

Subject Island: Case 1

(5) a. What do you think the speech interrupted *t*? Obj — Non Island

b. What do you think *t* interrupted the show? Subj — Non Island

c. What do you think the speech about global warming interrupted the show about *t*? Obj — Island

d. What do you think the speech about *t* interrupted the show about global warming? Subj — Island

Sprouse et al. (2012)			MG Parser
Subj. — Non Isl.	>	Obj. — Non Isl.	✓
Subj. — Non Isl.	>	Obj. — Isl.	✓
Subj. — Non Isl.	>	Subj. — Isl.	✓
Obj. — Non Isl.	>	Obj. — Isl.	✓
Obj. — Non Isl.	>	Subj. — Isl.	✓
Obj. — Isl.	>	Subj. — Isl.	×

Clause Type	MaxT	SumS
Obj./Non Island	14/*do*	19
Subj./Non Island	11/*do*	14
Obj./Island	23/*T2*	22
Subj./Island	15/*do*	20

Subject Island: Case 1

(5)
a. What do you think the speech interrupted *t*? — Obj — Non Island
b. What do you think *t* interrupted the show? — Subj — Non Island
c. What do you think the speech about global warming interrupted the show about *t*? — Obj — Island
d. * What do you think the speech about *t* interrupted the show about global warming? — Subj — Island

Sprouse et al. (2012)			MG Parser
Subj. — Non Isl.	>	Obj. — Non Isl.	✓
Subj. — Non Isl.	>	Obj. — Isl.	✓
Subj. — Non Isl.	>	Subj. — Isl.	✓
Obj. — Non Isl.	>	Obj. — Isl.	✓
Obj. — Non Isl.	>	Subj. — Isl.	✓
Obj. — Isl.	>	Subj. — Isl.	×

Clause Type	MaxT	SumS
Obj./Non Island	14/*do*	19
Subj./Non Island	11/*do*	14
Obj./Island	23/*T2*	22
Subj./Island	15/*do*	20

Subject Island: Case 2

(6)
a. Who *t* thinks the speech interrupted the primetime TV show?
Matrix — Non Island

b. What do you think *t* interrupted the primetime TV show?
Emb. — Non Island

c. Who *t* thinks the speech about global warming interrupted the primetime TV show?
Matrix — Island

d. What do you think the speech about *t* interrupted the primetime TV show?
Emb. — Island

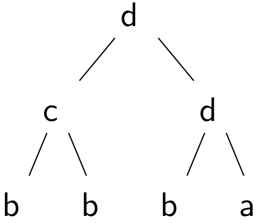
Sprouse et al. (2012)			MG Parser
Matrix — Non Isl.	>	Emb. — Non Isl.	✓
Matrix — Non Isl.	>	Matrix — Isl.	✓
Matrix — Non Isl.	>	Emb. — Isl.	✓
Matrix — Isl.	>	Emb. — Isl.	✓
Matrix — Isl.	>	Matrix — Isl.	✓
Emb. — Non Isl.	>	Emb. — Isl.	✓

Clause Type	MaxT	SumS
Matrix — Non Isl.	$5/C$	9
Emb. — Non Isl.	$11/do$	14
Matrix — Isl.	$11/T_{RC}$	9
Emb. — Isl.	$17/T_{RC}$	20

Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

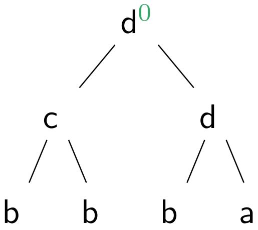
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

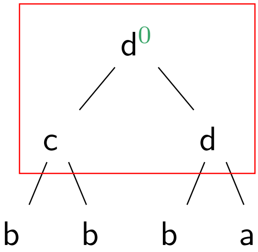
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

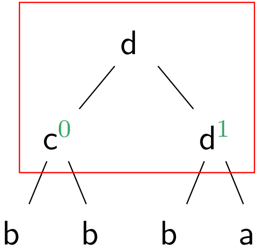
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

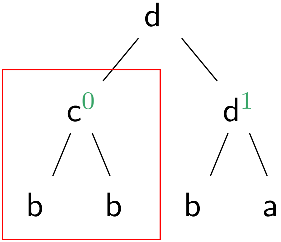
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

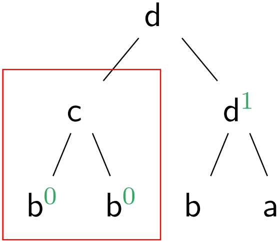
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

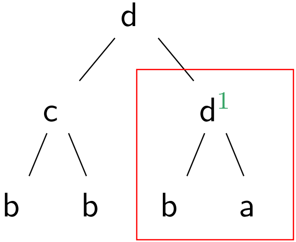
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

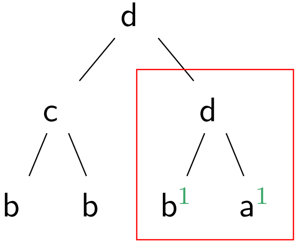
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

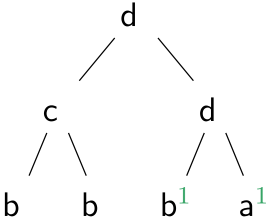
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

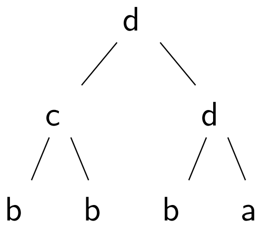
Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.

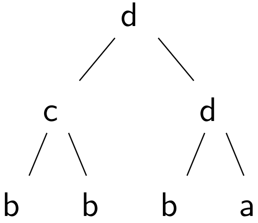


- Some island constrains arise naturally from this perspective (e.g., Adjunct Island Constraint, SpIC, ATB movement)

Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.

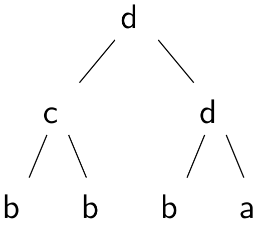


- Some island constrains arise naturally from this perspective (e.g., Adjunct Island Constraint, SpIC, ATB movement)
- Constraints improve parsing performance by **exponentially reducing** the search space (Stabler 2013)

Top-down Parsing + Grammaticalized Constraints?

Graf & De Santo (2019)

Sensing Tree Automata (Martens 2006) as a subregular bound on the complexity of syntactic dependencies.



- Some island constrains arise naturally from this perspective (e.g., Adjunct Island Constraint, SpIC, ATB movement)
- Constraints improve parsing performance by **exponentially reducing** the search space (Stabler 2013)
- Can be pre-compiled in the MG parse schema as a deterministic **top-down filter** (De Santo & Graf, in prep.)

Summary

Gradience from a categorical MG grammar?

- The **first** (quantitative) model of this kind!
- Overall, a success! ⇒ **just** from structural differences!
- Outlier is expected assuming grammaticalized constraints.

The tip of the iceberg!

- Modulate range of dependencies
- Other examples of gradience
- Cognitive vs. grammatical constraints? (Ferrara-Boston 2012)
- Syntactic constraints ∼ pruning the parsing space (Stabler 2013)
- Probing industrial-level language models (Wilcox et al. 2018; Torr et al. 2019)

Summary

Gradience from a categorical MG grammar?

- The **first** (quantitative) model of this kind!
- Overall, a success! ⇒ **just** from structural differences!
- Outlier is expected assuming grammaticalized constraints.

The tip of the iceberg!

- Modulate range of dependencies
- Other examples of gradience
- Cognitive vs. grammatical constraints? (Ferrara-Boston 2012)
- Syntactic constraints ∼ pruning the parsing space (Stabler 2013)
- Probing industrial-level language models (Wilcox et al. 2018; Torr et al. 2019)

Summary

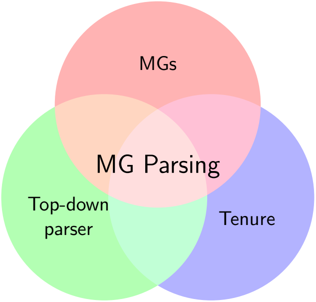
Gradience from a categorical MG grammar?

- The **first** (quantitative) model of this kind!
- Overall, a success! ⇒ **just** from structural differences!
- Outlier is expected assuming grammaticalized constraints.

The tip of the iceberg!

- Modulate range of dependencies
- Other examples of gradience
- Cognitive vs. grammatical constraints? (Ferrara-Boston 2012)
- Syntactic constraints ∼ pruning the parsing space (Stabler 2013)
- Probing industrial-level language models (Wilcox et al. 2018; Torr et al. 2019)

From the Trees (back) to the Forest

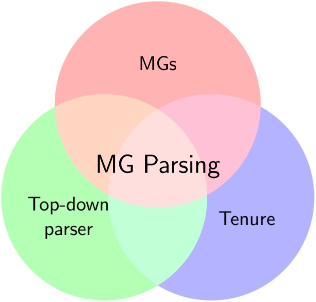


- Fully specified parsing model allows for precise predictions
- Tight connection with current generative syntax
- Successful on a variety of cross-linguistic constructions
- + insights about the structure of the grammar

Not Just Theoretical Insights!

- The human parser outperforms our fastest parsers

From the Trees (back) to the Forest

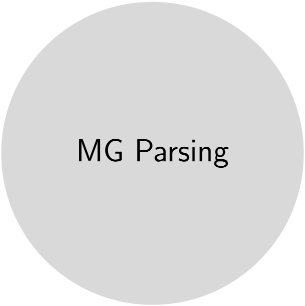


- Fully specified parsing model allows for precise predictions
- Tight connection with current generative syntax
- Successful on a variety of cross-linguistic constructions
- + insights about the structure of the grammar

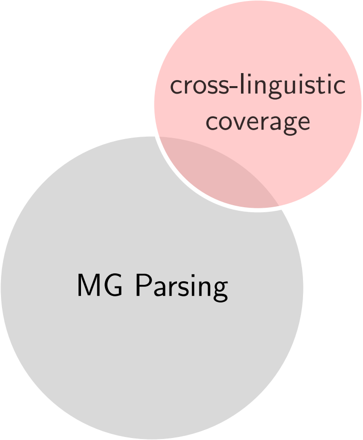
Not Just Theoretical Insights!

- The human parser outperforms our fastest parsers

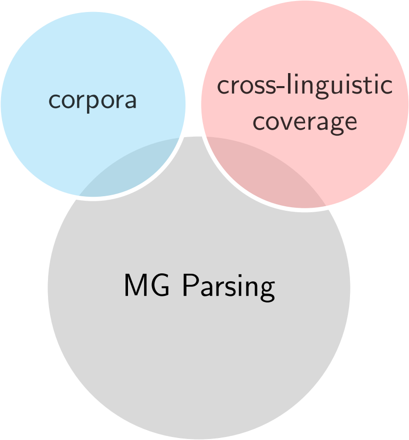
Looking Ahead: A Collaborative Enterprise



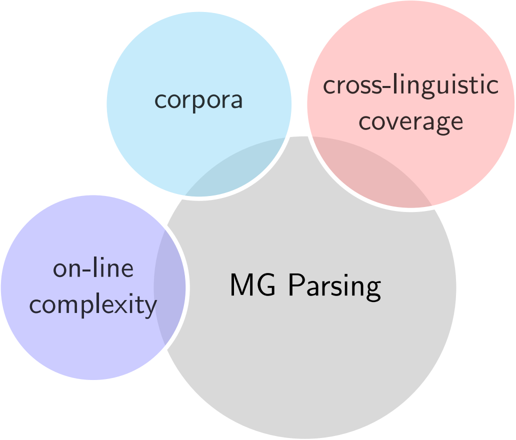
Looking Ahead: A Collaborative Enterprise



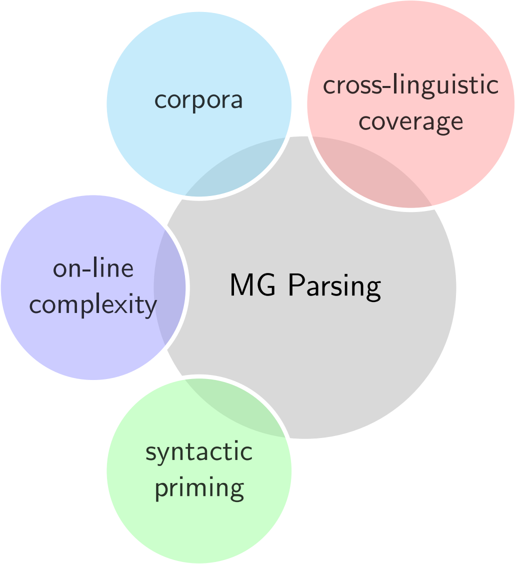
Looking Ahead: A Collaborative Enterprise



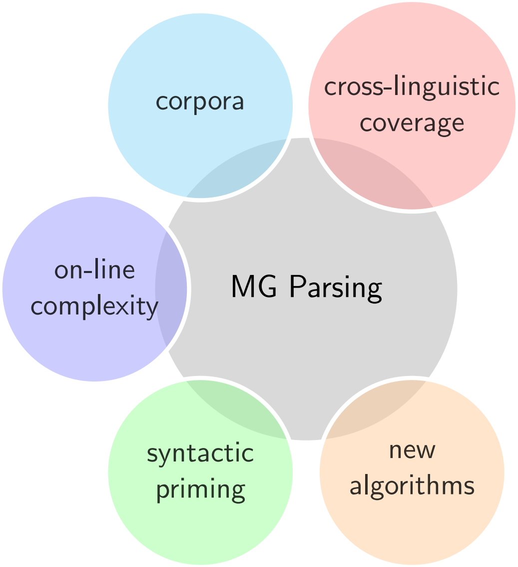
Looking Ahead: A Collaborative Enterprise



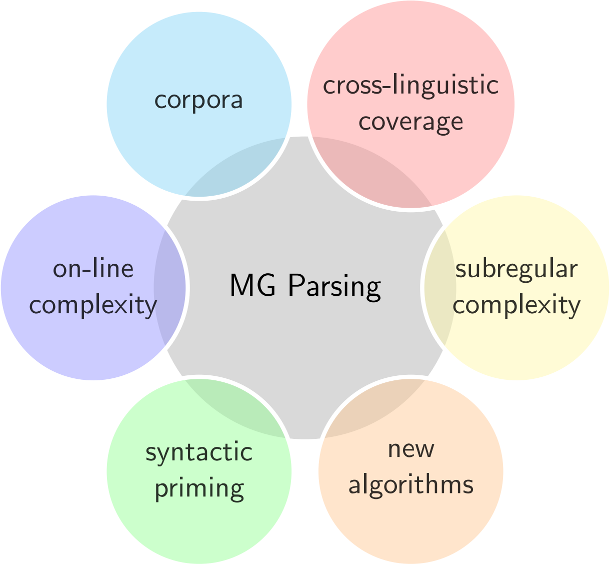
Looking Ahead: A Collaborative Enterprise



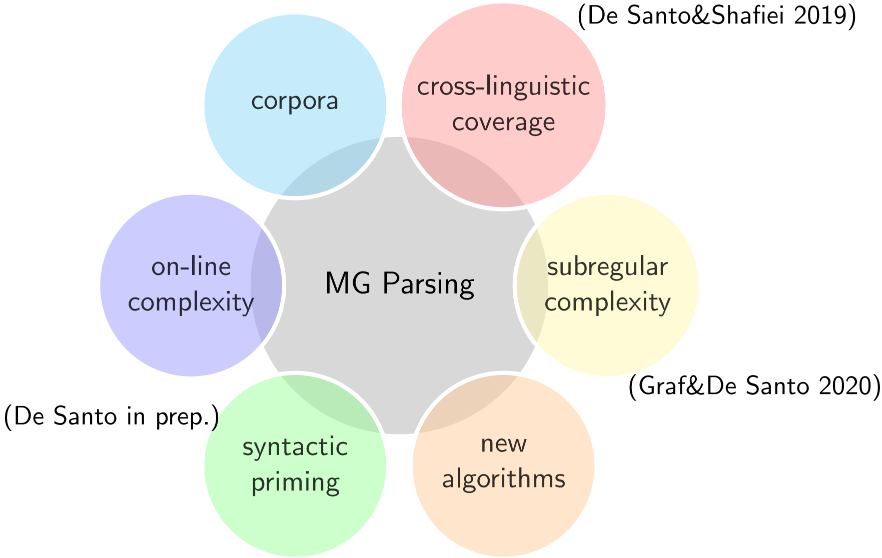
Looking Ahead: A Collaborative Enterprise



Looking Ahead: A Collaborative Enterprise



Looking Ahead: A Collaborative Enterprise



Looking Ahead: A Collaborative Enterprise

Selected References I

1. **Chomsky, N.** (1995). The minimalist program. Cambridge, Mass.: MIT Press.
2. **De Santo, A.** (2019). Testing a Minimalist gram- mar parser on Italian relative clause asymmetries. In *Proceedings of CMCL 2019*, June 6 2019, Minneapolis, Minnesota.
3. **De Santo, A.** (2020). MG Parsing as a Model of Gradient Acceptability in Syntactic Islands. (To appear) In *Proceedings of SCiL 2020*, Jan 2-5, New Orleans.
4. **De Santo, A.** and Shafiei, N. (2019). On the structure of relative clauses in Persian: Evidence from computational modeling and processing effects. *Talk at the NACIL2*, April 19-21 2019, University of Arizona.
5. **Graf, T.** and Monette, J. and Zhang, C. (2017). Relative Clauses as a Benchmark for Minimalist Parsing. *Journal of Language Modelling*.
6. **Kobele, G.M.**, Gerth S., and Hale. J. (2012). Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.
7. **Sprouse, J.**, Wagers, M. and Phillips, C. (2012). A test of the relation between working-memory capacity and syntactic island effects. *Language*.
8. **Stabler, E.P.** (2013). Bayesian, minimalist, incremental syntactic analysis. *Topics in Cognitive Science* 5:611–633.
9. **Stabler, E.P.** (1997). Derivational minimalism. In Logical aspects of computational linguistics, ed.

   Christian Retore, volume 1328 of Lecture Notes in Computer Science, 68–95. Berlin: Springer.

Appendix

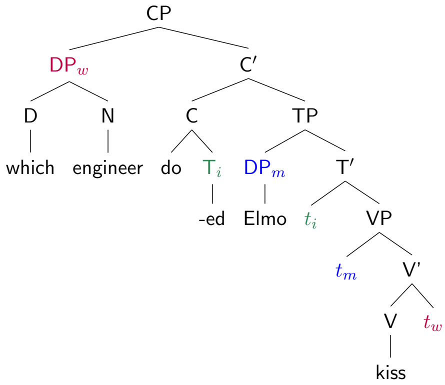
Why MGs?

1. Vast analytical coverage
   - MGs handle virtually all analyses in the generative literature
2. Centrality of derivation trees
   - MGs can be viewed as CFGs with a more complicated mapping from trees to strings
3. Simple parsing algorithms
   - Variant of a recursive descent parser for CFGs
     ⇒ cf. TAG (Rambow & Joshi, 1995; Demberg, 2008)

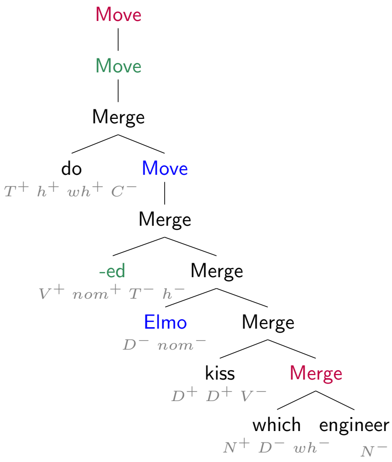
Some Important Properties of MGs

- MGs are weakly equivalent to MCFGs and thus mildly context-sensitive. (Harkema 2001, Michaelis 2001)
- But we can decompose them into two finite-state components: (Michaelis et al. 2001, Kobele et al. 2007, Monnich 2006)
  - a regular language of well-formed derivation trees
  - an MSO-definable mapping from derivations to phrase structure trees
- **Remember:** Every regular tree language can be re-encoded as a CFG (with more fine-grained non-terminal labels). (Thatcher 1967)

Fully Specified Derivation Trees



Phrase Structure Tree



Derivation Tree

Technical Fertility of MGs

MGs can accommodate the full syntactic toolbox:

- sidewards movement (Stabler, 2006; Graf 2013)
- affix hopping (Graf 2012; Graf2013)
- clustering movement (Gartner & Michaelis 2010)
- tucking in (Graf 2013)
- ATB movement (Kobele 2008)
- copy movement (Kobele 2006)
- extraposition (Hunter &Frank 2014)
- Late Merge (Kobele 2010; Graf 2014)
- Agree (Kobele 2011; Graf 2011)
- adjunction (Fowlie 2013; Hunter 2015)
- TAG-style adjunction (Graf 2012)

Why These Metrics?

- These complexity metrics are all related to **storage cost** (cf. Gibson, 1998)
- We could implement alternative ones (cf. Ferrara-Boston, 2012)
  - number of bounding nodes / phases
  - surprisal
  - feature intervention
  - status of discourse referents
  - integration, retrieval, ...
- We want to keep the model **simple** (but not **trivial**):
  - Tenure and Size only refer to the geometry of the derivation
  - they are sensitive the specifics of tree-traversal (cf. node-count; Hale, 2001)

Why These Metrics?

- These complexity metrics are all related to **storage cost** (cf. Gibson, 1998)
- We could implement alternative ones (cf. Ferrara-Boston, 2012)
  - number of bounding nodes / phases
  - surprisal
  - feature intervention
  - status of discourse referents
  - integration, retrieval, ...
- We want to keep the model **simple** (but not **trivial**):
  - Tenure and Size only refer to the geometry of the derivation
  - they are sensitive the specifics of tree-traversal (cf. node-count; Hale, 2001)

Italian Subjects: Probing the Results

Clause Type	MaxT	SumS
obj. SRC	8/*che*	18
obj. ORC	11/*ha*	24
obj. ORCp	16/*Foc*	31
subj. SRC	21/v'	37
subj. ORC	21/v'	44
subj. ORCp	28/v'	56
matrix SVO	3/*ha*/v'	7
matrix VOS	7/*Top*/*Foc*	11
VS unacc	2/vP	3
VS unerg	7/*Top*/*Foc*	11

Table: Summary of MaxT (*value*/*node*) and SumS by construction. Obj. and subj. indicate the landing site of the RC head in the matrix clause.

Postverbal Asymmetries: Possible Accounts?

SRC > ORC

- DLT, active-filler strategy, Competition model, ...

ORC > ORCp

- more problematic (e.g., for DLT)
- can be explained by
  1. economy of gap prediction + structural re-analysis;
  2. intervention effects + featural Relativized Minimality

Can we give a purely structural account?

Postverbal Asymmetries: Possible Accounts?

SRC > ORC

- DLT, active-filler strategy, Competition model, ...

ORC > ORCp

- more problematic (e.g., for DLT)
- can be explained by
  1. economy of gap prediction + structural re-analysis;
  2. intervention effects + featural Relativized Minimality

Can we give a purely structural account?

Postverbal Asymmetries: Possible Accounts?

SRC > ORC

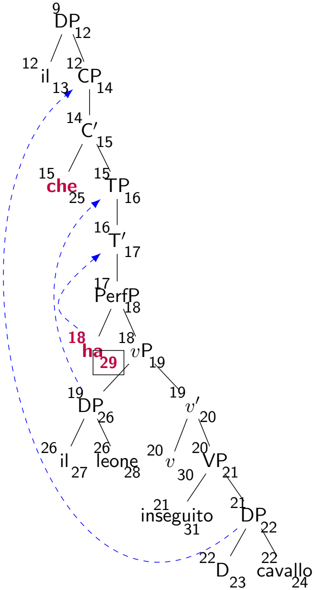
- DLT, active-filler strategy, Competition model, ...

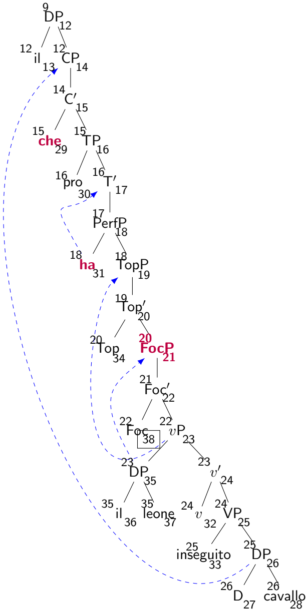
ORC > ORCp

- more problematic (e.g., for DLT)
- can be explained by
  1. economy of gap prediction + structural re-analysis;
  2. intervention effects + featural Relativized Minimality

Can we give a purely structural account?

Results: ORC > ORCp





Additional Constructions

- Ambiguity in Matrix Clauses

(7) Ha chiamato Gio
Has called Giovanni
a. "He/she/it called Gio" **SVO**
b. "Gio called" **VS**

- Unaccusatives vs. Unergatives

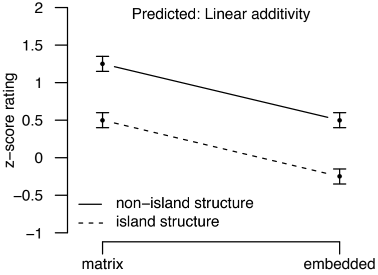
(8) È arrivato Gio
Is arrived Gio
"Gio arrived" **Unaccusative**

(9) Ha corso Gio
Has ran Gio
"Gio ran" **Unergative**

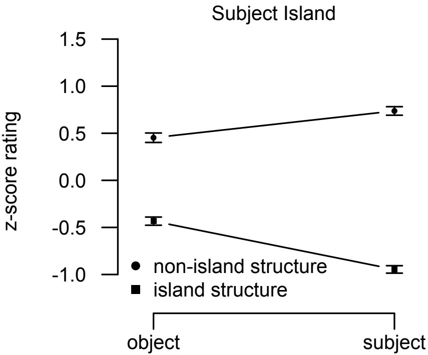
Gradience in Islands

A factorial design for islands effect:

- Gap Position × Structure



Deriving Pairwise Comparisons



- Subj — Non Island > Obj — Non Island
- Subj — Non Island > Obj — Island
- Subj — Non Island > Subj — Island
- etc.

A Caveat on Island Effects

The Goal

Can gradience in acceptability judgments arise from a categorical grammar due to processing factors?

- Sprouse et al.'s (2012) design is ideal for the MG model.

But I am not interested in island effects per se:

- Islands: grammatical or processing effects? (Hofmeister et al., 2012a; Sprouse et al., 2012a,b)
  - hence, not modeling super-additivity
  - spoilers: maybe we get some insights?
- Islands: syntax or semantics? (Truswell, 2011; Kush et al., 2018; Matchin et al., 2018)

A Caveat on Island Effects

The Goal

Can gradience in acceptability judgments arise from a categorical grammar due to processing factors?

- Sprouse et al.'s (2012) design is ideal for the MG model.

But I am not interested in island effects per se:

- Islands: grammatical or processing effects? (Hofmeister et al., 2012a; Sprouse et al., 2012a,b)
  - hence, not modeling super-additivity
  - spoilers: maybe we get some insights?
- Islands: syntax or semantics? (Truswell, 2011; Kush et al., 2018; Matchin et al., 2018)

A Caveat on Island Effects

The Goal

Can gradience in acceptability judgments arise from a categorical grammar due to processing factors?

- Sprouse et al.'s (2012) design is ideal for the MG model.

But I am not interested in island effects per se:

- Islands: grammatical or processing effects? (Hofmeister et al., 2012a; Sprouse et al., 2012a,b)
  - hence, not modeling super-additivity
  - spoilers: maybe we get some insights?
- Islands: syntax or semantics? (Truswell, 2011; Kush et al., 2018; Matchin et al., 2018)

A Caveat on Island Effects

The Goal

Can gradience in acceptability judgments arise from a categorical grammar due to processing factors?

- Sprouse et al.'s (2012) design is ideal for the MG model.

But I am not interested in island effects per se:

- Islands: grammatical or processing effects? (Hofmeister et al., 2012a; Sprouse et al., 2012a,b)
  - hence, not modeling super-additivity
  - spoilers: maybe we get some insights?
- Islands: syntax or semantics? (Truswell, 2011; Kush et al., 2018; Matchin et al., 2018)

Subject Islands

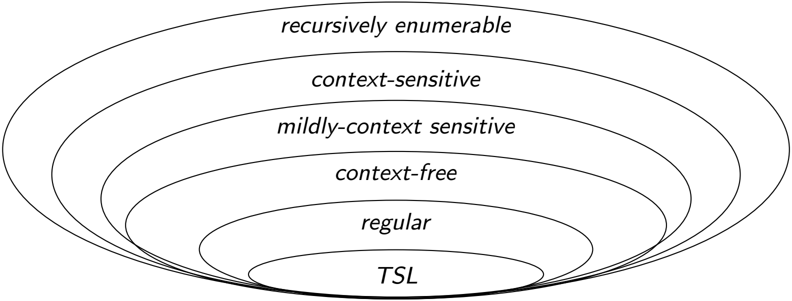
Case 1:

(10)
a. What do you think the speech interrupted *t*? — Obj — Non Island
b. What do you think *t* interrupted the show? — Subj — Non Island
c. What do you think the speech about global warming interrupted the show about *t*? — Obj — Island
d. What do you think the speech about *t* interrupted the show about global warming? — Subj — Island

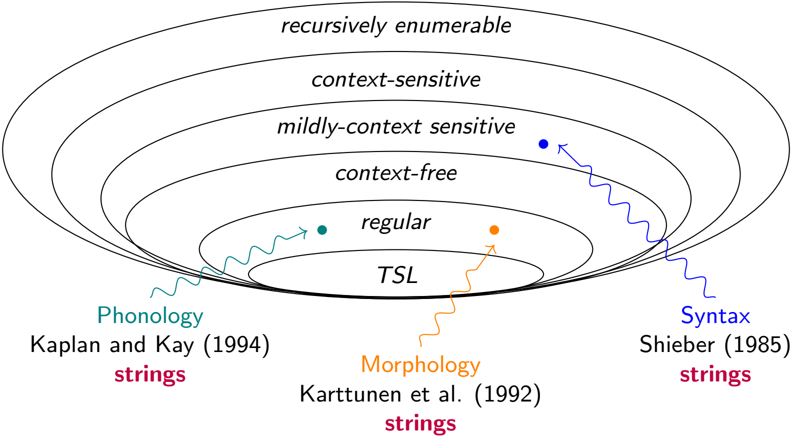
Case 2:

(11)
a. Who *t* thinks the speech interrupted the primetime TV show? — Matrix — Non Island
b. What do you think *t* interrupted the primetime TV show? — Emb. — Non Island
c. Who *t* thinks the speech about global warming interrupted the primetime TV show? — Matrix — Island
d. What do you think the speech about *t* interrupted the primetime TV show? — Emb. — Island

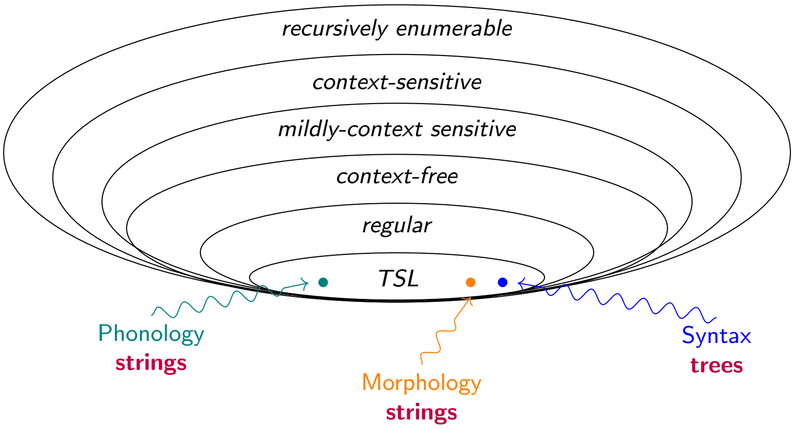
Subregular Complexity



Subregular Complexity



Subregular Complexity



Cognitive Parallelism

Strong Cognitive Parallelism Hypothesis

Phonology, (morphology), and syntax have the **same subregular complexity** over their respective **structural representations**.

We gain a unified perspective on:

- typology
- learnability
- cognition

Cognitive Parallelism

Strong Cognitive Parallelism Hypothesis

Phonology, (morphology), and syntax have the **same subregular complexity** over their respective **structural representations**.

We gain a unified perspective on:

- typology
  - × Intervocalic Voicing iff applied an even times in the string
  - × Have a CP iff it dominates ≥ 3 TPs
- learnability
- cognition

Cognitive Parallelism

Strong Cognitive Parallelism Hypothesis

Phonology, (morphology), and syntax have the **same subregular complexity** over their respective **structural representations**.

We gain a unified perspective on:

- typology
  - × Intervocalic Voicing iff applied an even times in the string
  - × Have a CP iff it dominates ≥ 3 TPs
- learnability

  Learnable from positive examples of strings/trees.
- cognition

Cognitive Parallelism

Strong Cognitive Parallelism Hypothesis

Phonology, (morphology), and syntax have the **same subregular complexity** over their respective **structural representations**.

We gain a unified perspective on:

- typology
  - × Intervocalic Voicing iff applied an even times in the string
  - × Have a CP iff it dominates ≥ 3 TPs
- learnability
  Learnable from positive examples of strings/trees.
- cognition
  Finite, flat memory