Constraints Emerge from Merge

Thomas Graf
tgraf@ucla.edu
tgraf.bol.ucla.edu

University of California, Los Angeles

Apr 26, 2013

Take-Home Message

Questions

- How do constraints fit into syntax?
- What are their properties?

This Talk

- Constraints are mediated by operations.
- A computational perspective makes this connection explicit.
- Linking constraints to operations limits their power and makes new empirical predictions.

Take-Home Message

Questions

- How do constraints fit into syntax?
- What are their properties?

This Talk

- Constraints are mediated by operations.
- A computational perspective makes this connection explicit.
- Linking constraints to operations limits their power
  and makes new empirical predictions.

Outline

1. Formal Perspective
   - Background: Minimalist Grammars
   - Constraints as Logical Formulas
   - Main Result: Constraints $\equiv$ Merge
2. Are Merge-Definable Constraints Enough? A Look at Binding
   - Computing Principle B
   - English
   - American Sign Language (ASL)
3. Strong Islands as Calculation Errors
   - Adjunct Languages
   - Deriving Adjunct Islands and Parasitic Gaps
   - Some Speculative Extensions
4. Conclusion & Outlook

Outline

1. Formal Perspective
   - Background: Minimalist Grammars
   - Constraints as Logical Formulas
   - Main Result: Constraints $\equiv$ Merge
2. Are Merge-Definable Constraints Enough? A Look at Binding
   - Computing Principle B
   - English
   - American Sign Language (ASL)
3. Strong Islands as Calculation Errors
   - Adjunct Languages
   - Deriving Adjunct Islands and Parasitic Gaps
   - Some Speculative Extensions
4. Conclusion & Outlook

Minimalist Grammars (MGs)

- formalization of Minimalist syntax without Agree/phases (Stabler 1997)
- many extensions to make them more faithful (Frey and Gärtner 2002; Graf 2012; Kobele 2002, 2012; Stabler 2003, 2006, 2011, among others)
- original version suffices for our purposes

Core Idea of MGs

- Operations: **Merge** and **Move**
- lexical items annotated with features
- features come in two polarities
- each operation must check two features of opposite polarity

Minimalist Grammars (MGs)

- formalization of Minimalist syntax without Agree/phases (Stabler 1997)
- many extensions to make them more faithful (Frey and Gärtner 2002; Graf 2012; Kobele 2002, 2012; Stabler 2003, 2006, 2011, among others)
- original version suffices for our purposes

Core Idea of MGs

- Operations: **Merge** and **Move**
- lexical items annotated with features
- features come in two polarities
- each operation must check two features of opposite polarity

Merge: Example 1

Assembling [$_{\text{DP}}$ the men]

the men

- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

Assembling [$_{\text{DP}}$ the men]

the men

- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

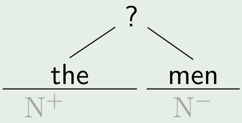
Assembling [$_{\text{DP}}$ the men]

$$\frac{\text{the}}{N^{+}} \quad \frac{\text{men}}{N^{-}}$$

- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

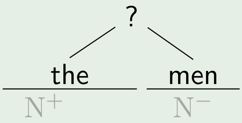
Assembling [$_{\text{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

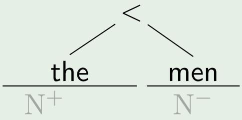
Assembling [$_{\mathrm{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^{-}, D^{-}, V^{-}, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

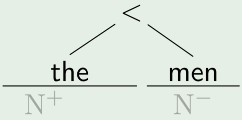
Assembling [$_{\text{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

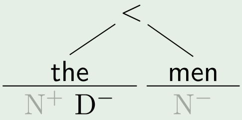
Assembling [$_{\text{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

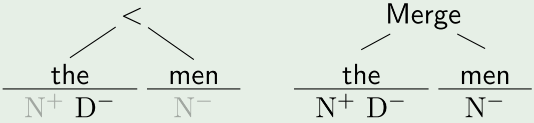
Assembling [$_{\text{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

Assembling [$_{\text{DP}}$ the men]



- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 1

Assembling [$_{\text{DP}}$ the men]

$<$

the — N^+ D^- | men — N^-

Merge[N]

the — N^+ D^- | men — N^-

- Merge triggered by features of opposite polarities
- Label points to branch leading to projecting head
- Head must have a category feature (N^-, D^-, V^-, ...)
- Derivation tree differs only with respect to labels

Merge: Example 2

Assembling [$_{VP}$ the men like which men]

the men like which men

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{VP}$ the men like which men]

the men like which men

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{VP}$ the men like which men]

the	men	like	which	men
$N^+ \; D^-$	N^-			

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

Assembling [$_{\mathrm{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

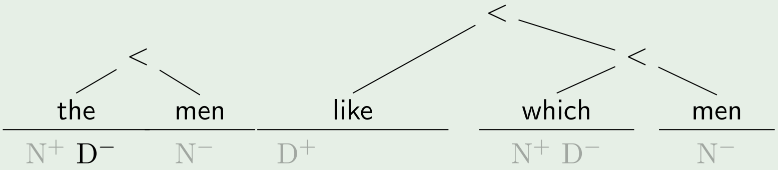
Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]

- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

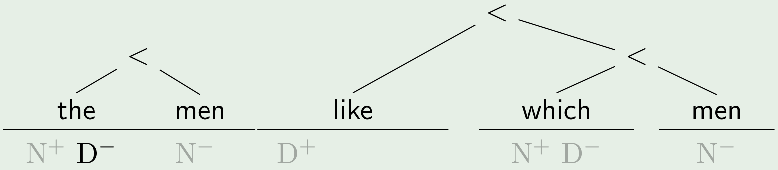
Assembling $[_{\text{VP}}$ the men like which men$]$



- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

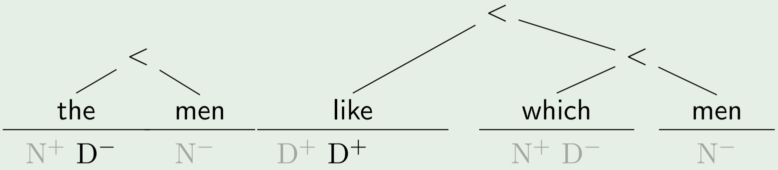
Assembling [$_{\text{VP}}$ the men like which men]



- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

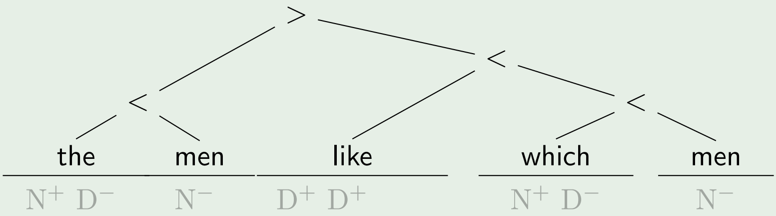
Assembling [$_{\text{VP}}$ the men like which men]



- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

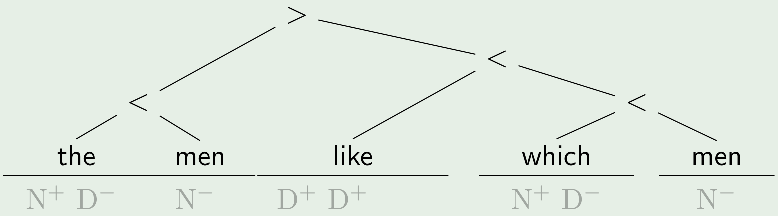
Assembling [$_{\text{VP}}$ the men like which men]



- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2

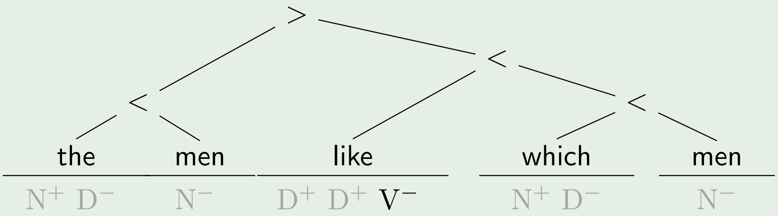
Assembling [$_{\text{VP}}$ the men like which men]



- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

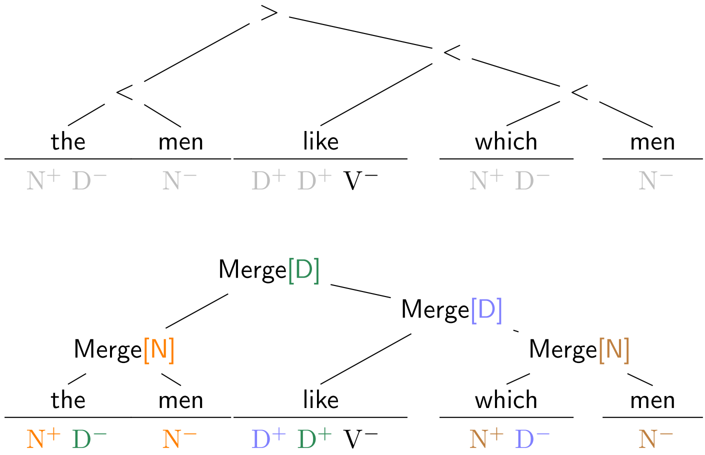
Merge: Example 2

Assembling [$_{\text{VP}}$ the men like which men]



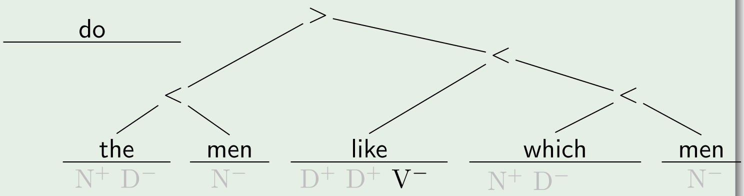
- *the* and *men* merged as before
- same steps for *which men*
- *like* selects *which men*
- *like* selects *the men*
- *like* needs a category feature

Merge: Example 2 [cont.]



Move (Multi-Dominance Implementation)

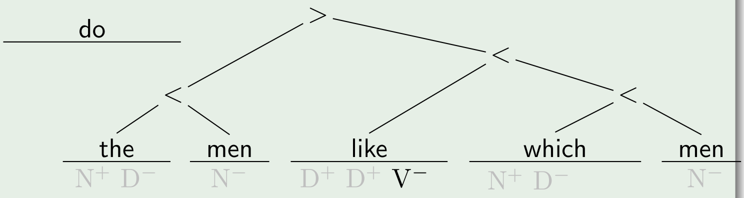
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

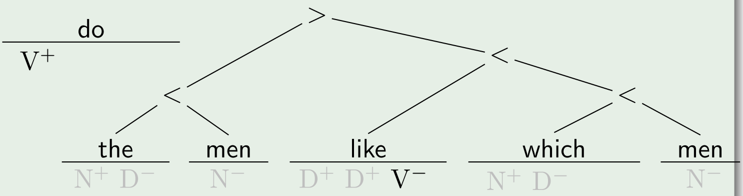
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

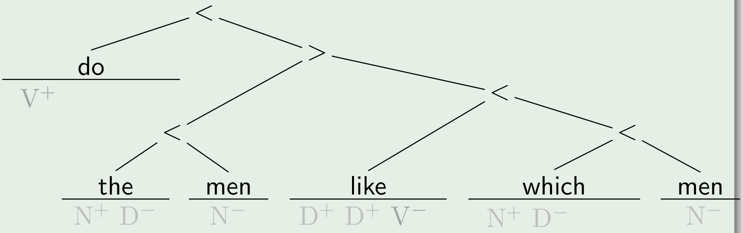
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

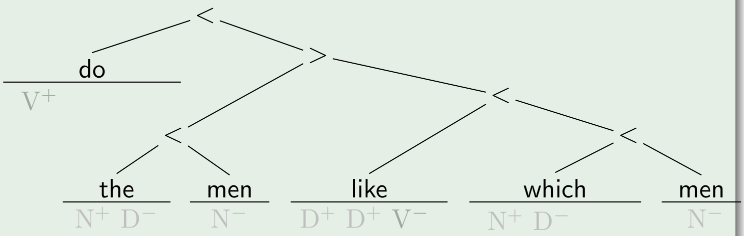
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

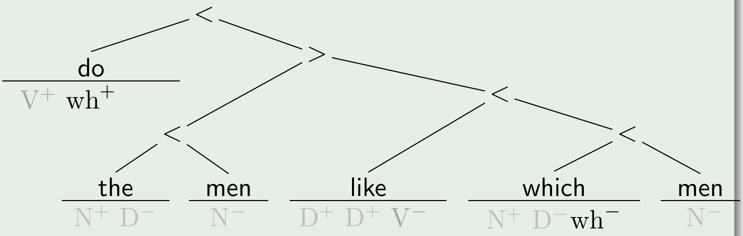
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

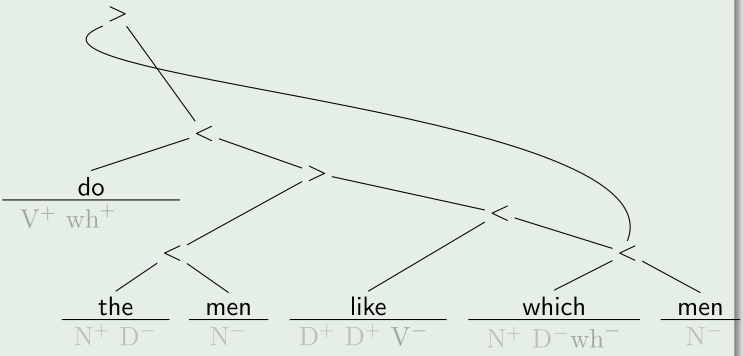
Assembling “which men do the men like?”



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

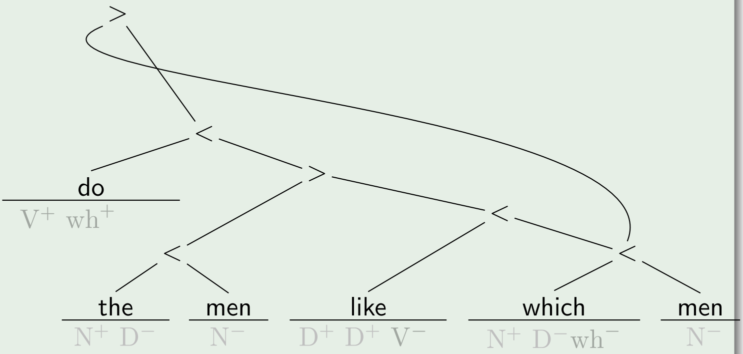
Assembling “which men do the men like?”



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Move (Multi-Dominance Implementation)

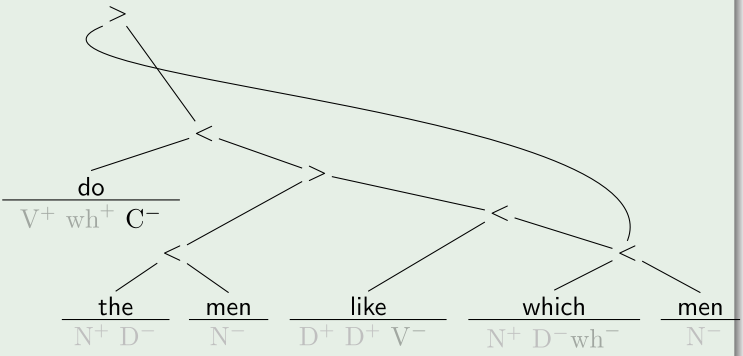
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

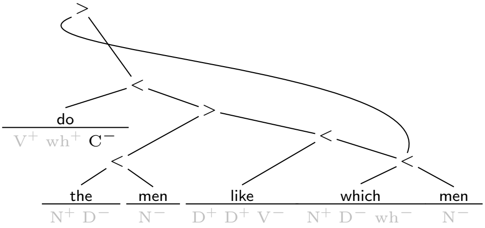
Move (Multi-Dominance Implementation)

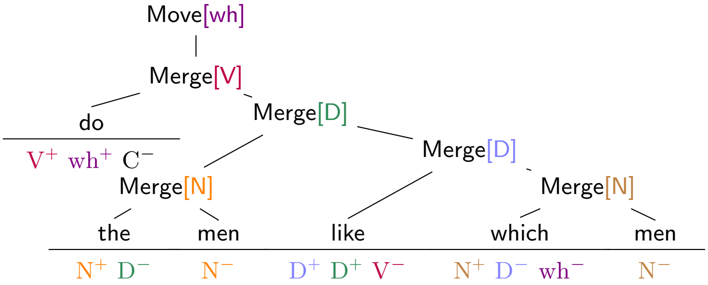
Assembling "which men do the men like?"



- Merge *do*
- Move triggered by features of opposite polarity
- *do* must have a category feature

Derivation Trees with Move





MG Summary

An MG is given by a **set of feature-annotated lexical items**.
It generates all (multi-dominance) trees that are CPs built from the available lexical items.

Example

$\frac{\text{men}}{\mathrm{N}^-}$	$\frac{\text{the}}{\mathrm{D}^+\ \mathrm{N}^-}$	$\frac{\text{which}}{\mathrm{D}^+\ \mathrm{N}^-\ \mathrm{wh}^-}$
$\frac{\text{like}}{\mathrm{D}^+\ \mathrm{D}^+\ \mathrm{V}^-}$	$\frac{\text{do}}{\mathrm{V}^+\ \mathrm{wh}^+\ \mathrm{C}^-}$	$\frac{\varepsilon}{\mathrm{V}^+\ \mathrm{C}^-}$

Generated sentences: The men like the men.
Which men do the men like.
Which men do like the men.

Formalizing Constraints

Constraint statement c that must be satisfied in order for a tree to be well-formed

Logical Formula statement ϕ that must be satisfied in order for a structure to be a model of ϕ

$\Rightarrow$ **Constraints $\equiv$ Logical Formulas**

(Kracht 1995; Rogers 1998; Potts 2001; Pullum 2007)

Monadic Second-Order for Trees (MSO)

- standard logical connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$)
- quantification over nodes ($\forall x$, $\exists x$)
- quantification over sets of nodes ($\forall X$, $\exists X$)
- predicate for each node label (him(x), DP(x))
- dominance relation ($\triangleleft$)
- any predicate definable from the above

Formalizing Constraints

Constraint statement c that must be satisfied in order for a tree to be well-formed

Logical Formula statement ϕ that must be satisfied in order for a structure to be a model of ϕ

$\Rightarrow$ **Constraints $\equiv$ Logical Formulas**

(Kracht 1995; Rogers 1998; Potts 2001; Pullum 2007)

Monadic Second-Order for Trees (MSO)

- standard logical connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$)
- quantification over nodes ($\forall x$, $\exists x$)
- quantification over sets of nodes ($\forall X$, $\exists X$)
- predicate for each node label (him(x), DP(x))
- dominance relation ($\triangleleft$)
- any predicate definable from the above

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z]\big]\Big]$$

"For every x

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x\Big[\text{anaphor}(x) \rightarrow \exists y\big[\text{c-com}(y,x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z,x) \wedge y \in Z]\big]\Big]$$

"For every x that is an anaphor

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y,x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z,x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP,

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{bind-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x and

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x\Big[\text{anaphor}(x) \rightarrow \exists y\big[\text{c-com}(y,x) \wedge \text{DP}(y) \wedge \exists Z[\text{bind-dom}(Z,x) \wedge y \in Z]\big]\Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x and contains y."

The Power of MSO

MSO can state conditions on both nodes and domains
$\Rightarrow$ powerful enough for **almost all syntactic constraints**

- constraints over phrase structure trees,
- constraints over derivations,
- transderivational constraints/economy conditions.

Undefinable Constraints

- "Subtrees A and B are identical"
  $\Rightarrow$ problematic for ellipsis as deletion under (syntactic) identity
- "The meaning of subtree A implies the meaning of B"
  $\Rightarrow$ problematic for semantic constraints in syntax

The Power of MSO

MSO can state conditions on both nodes and domains
$\Rightarrow$ powerful enough for **almost all syntactic constraints**

- constraints over phrase structure trees,
- constraints over derivations,
- transderivational constraints/economy conditions.

Undefinable Constraints

- "Subtrees A and B are identical"
  $\Rightarrow$ problematic for ellipsis as deletion under (syntactic) identity
- "The meaning of subtree A implies the meaning of B"
  $\Rightarrow$ problematic for semantic constraints in syntax

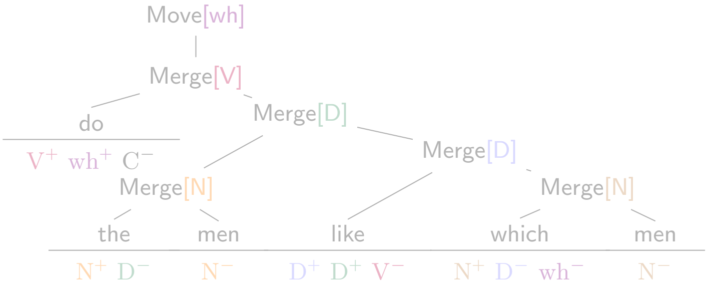
Computing MSO-Constraints

MSO & Tree Automata (Thatcher and Wright 1968; Doner 1970)

A constraint is MSO-definable iff it can be computed by a **(finite-state) tree automaton**.

A tree automaton

- assigns each node in a tree one of finitely many **states**, and
- accepts the tree iff its root is assigned a **final state**.



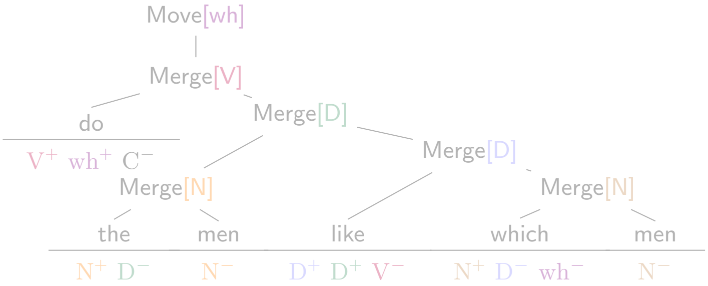
Computing MSO-Constraints

MSO & Tree Automata (Thatcher and Wright 1968; Doner 1970)

A constraint is MSO-definable iff it can be computed by a **(finite-state) tree automaton**.

A tree automaton

- assigns each node in a tree one of finitely many **states**, and
- accepts the tree iff its root is assigned a **final state**.



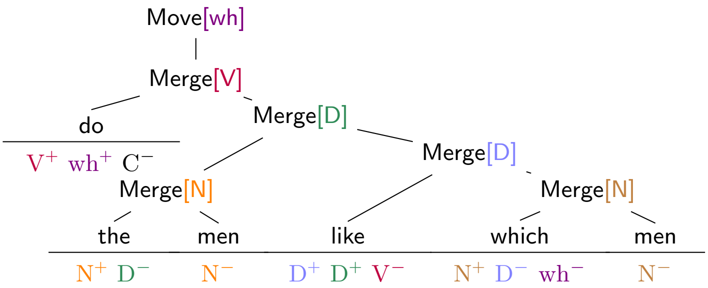
Computing MSO-Constraints

MSO & Tree Automata (Thatcher and Wright 1968; Doner 1970)

A constraint is MSO-definable iff it can be computed by a **(finite-state) tree automaton**.

A tree automaton

- assigns each node in a tree one of finitely many **states**, and
- accepts the tree iff its root is assigned a **final state**.



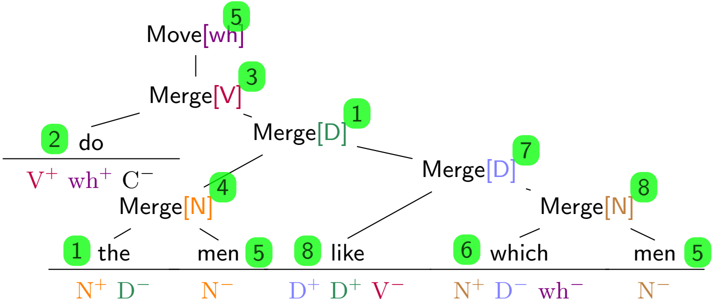
Computing MSO-Constraints

MSO & Tree Automata (Thatcher and Wright 1968; Doner 1970)

A constraint is MSO-definable iff it can be computed by a **(finite-state) tree automaton**.

A tree automaton

- assigns each node in a tree one of finitely many **states**, and
- accepts the tree iff its root is assigned a **final state**.



Interim Summary

Minimalist Syntax

- Formalized in terms of MGs
- Operations: Merge and Move (Agree omitted for convenience)
- Triggered by **checking features of opposite polarities**

Constraints

- Constraints $\equiv$ MSO formulas $\equiv$ tree automata
- MSO can talk about both nodes and sets of nodes
  $\Rightarrow$ **expressive enough for syntax**
- Tree automata compute constraints in a local way
  using finitely bounded number of states ($\approx$ working memory)
  $\Rightarrow$ **cognitive plausibility**

The Central Result

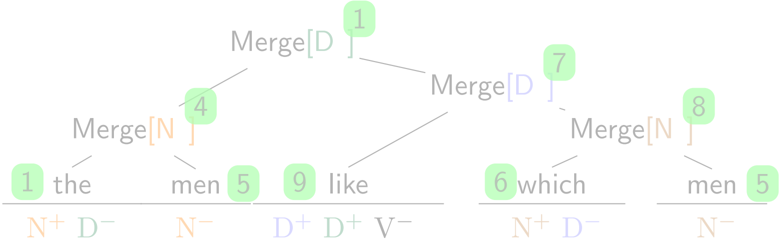
MSO-Constraints ≡ Merge (Graf 2011; Kobele 2011)

A constraint C can be expressed via Merge iff C is MSO-definable.

Proof idea

- convert constraint C into tree automaton A
- incorporate states of A into feature calculus

  $\Rightarrow$ “refined” grammar **expresses C via Merge**



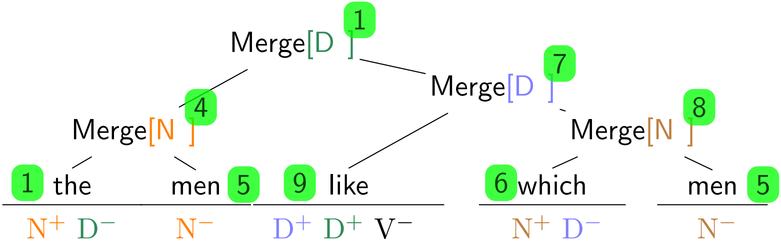
The Central Result

MSO-Constraints ≡ Merge (Graf 2011; Kobele 2011)

A constraint C can be expressed via Merge iff C is MSO-definable.

Proof idea

- convert constraint C into tree automaton A
- incorporate states of A into feature calculus
  $\Rightarrow$ "refined" grammar **expresses C via Merge**



The Central Result

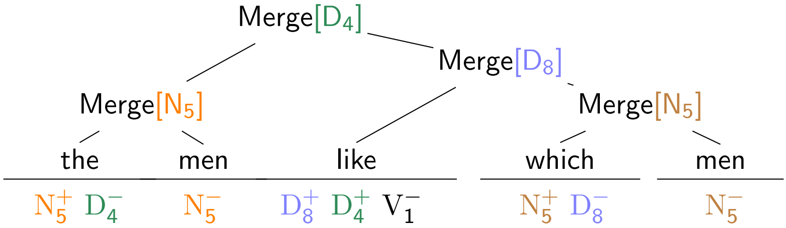
MSO-Constraints ≡ Merge (Graf 2011; Kobele 2011)

A constraint C can be expressed via Merge iff C is MSO-definable.

Proof idea

- convert constraint C into tree automaton A
- incorporate states of A into feature calculus

  $\Rightarrow$ "refined" grammar **expresses C via Merge**



Recap: What Just Happened?

- Monadic Second-Order logic as description language:
  - powerful enough for stating syntactic constraints
  - computable with finite working-memory
- Every MSO-constraint expressible purely through Merge
  Metaphor: Put memory states into category features

The New Perspective on Constraints

- Existence of MSO-definable constraints unsurprising given power of Merge
- But are there really only MSO-constraints in syntax?
  What would the implications be?

Recap: What Just Happened?

- Monadic Second-Order logic as description language:
  - powerful enough for stating syntactic constraints
  - computable with finite working-memory
- Every MSO-constraint expressible purely through Merge
  Metaphor: Put memory states into category features

The New Perspective on Constraints

- Existence of MSO-definable constraints unsurprising given power of Merge
- But are there really only MSO-constraints in syntax?
  What would the implications be?

Outline

Technical Assumptions

- **Assumption 1: Syntactic Binding $\neq$ Discourse Binding**
  only syntactic binding (him_s), not discourse binding (him_d)
  $\Rightarrow$ every pronoun supposedly needs a syntactic antecedent
- **Assumption 2: (Strongly) Index-Free Binding**
  there are no indices in syntax
  $\Rightarrow$ syntax tests for availability of **some** grammatical reading, does not evaluate grammaticality of **specific** readings

Example

(1) a. Every patient said that he_d should sedate him_s.
b. * Every patient said that he_s should sedate him_s.
c. Every patient told some doctor that he_s should sedate him_s.

Technical Assumptions

- **Assumption 1: Syntactic Binding $\neq$ Discourse Binding**
  only syntactic binding (him$_s$), not discourse binding (him$_d$)
  $\Rightarrow$ every pronoun supposedly needs a syntactic antecedent
- **Assumption 2: (Strongly) Index-Free Binding**
  there are no indices in syntax
  $\Rightarrow$ syntax tests for availability of **some** grammatical reading, does not evaluate grammaticality of **specific** readings

Example

(1)
a. Every patient said that he$_d$ should sedate him$_s$.
b. * Every patient said that he$_s$ should sedate him$_s$.
c. Every patient told some doctor that he$_s$ should sedate him$_s$.

Principle B: Limited Obviation

Principle B is difficult because of its **obviation requirement** (= no local binding).

Syntactic Binding and Merge

Syntactic Binding is Merge-definable iff **Limited Obviation** holds.

Limited Obviation

For every binding domain, its syntactically bound pronouns need at most a total of n antecedents to yield a grammatical reading.

So, what does that mean?

If a binding domain contains more than n bound pronouns, those additional pronouns can be coreferent with pronouns in the same domain ⇒ Principle B exceptions

Principle B: Limited Obviation

Principle B is difficult because of its **obviation requirement** (= no local binding).

Syntactic Binding and Merge

Syntactic Binding is Merge-definable iff **Limited Obviation** holds.

Limited Obviation

For every binding domain, its syntactically bound pronouns need at most a total of n antecedents to yield a grammatical reading.

So, what does that mean?

If a binding domain contains more than n bound pronouns, those additional pronouns can be coreferent with pronouns in the same domain ⇒ Principle B exceptions

Principle B: Limited Obviation

Principle B is difficult because of its **obviation requirement** (= no local binding).

Syntactic Binding and Merge

Syntactic Binding is Merge-definable iff **Limited Obviation** holds.

Limited Obviation

For every binding domain, its syntactically bound pronouns need at most a total of n antecedents to yield a grammatical reading.

So, what does that mean?

If a binding domain contains more than n bound pronouns, those additional pronouns can be coreferent with pronouns in the same domain $\Rightarrow$ Principle B exceptions

How would one Falsify Limited Obviation?

- All binding proposals agree that there is some domain within which pronouns may not be syntactically bound $\approx$ binding/obviation domain
- binding domain $\leq$ CP
- Within a single CP, there are three ways of introducing an unbounded number of pronominal DPs:
  - adjuncts
  - nested TPs/*v*Ps, VPs, and DPs
  - coordination
- **Limited Obviation** is **violated only if the pronouns in these configurations all obviate each other** (i.e. are mandatorily disjoint in reference).

Adjuncts

Pronouns contained by adjuncts usually **lack obviation**.

(2) Every/No/Some woman put the box down in front of her.

But even when obviation can be observed, pronouns contained by distinct adjuncts do not obviate each other.

(3) a. * Every/No/Some priest sacrificed a goat for him.

b. Every/No/Some Egyptian goddess asked of some priest to sacrifice a goat for her in honor of her.

Hence adjuncts increase the required number of antecedents only by a limited amount.

Recursion Patterns

- TPs/*v*Ps introduce new binding domains

(4) a. * Every/No/Some patient said that he wants him to sedate him.

b. Every/No/Some patient told some doctor that he believes him (to want him (to believe him ...)) to sedate him.

- DPs introduce new binding domains or allow local binding

(5) a. Every/No/Some movie actor complained about [the Sun's article on [him and some paparazzi's picture of him]].

b. Every/No/Some client wanted to see a [presentation of [a presentation to him] to him].

Coordination

Coordination involving bound pronouns is **ungrammatical if the two pronouns are identical**.

(6) a. Every/No/Some football player told every/no/some cheerleader that the coach wants to see him and her in the office.

b. * Every/No/Some football player told every/no/some masseur that the coach wants to see him and him in the office.

Since every language has only a finite number of distinct pronouns, coordination can only introduce a bounded number of pronouns that obviate each other.

A Counterexample in ASL? (Graf and Abner 2012)

Coordination of bound pronouns is grammatical in ASL.

(7) ALL$_{i}$ WRESTLER$_{i}$ INFORM$_{j}$ SOMEONE$_{j}$ SWIMMER$_{j}$ THAT IX$_{i/j}$ IX$_{j/i}$ WILL RIDE-IN-VEHICLE LIMO GO-TO DANCE
Every wrestler$_i$ told some swimmer$_j$ that him$_{i/j}$ and him$_{j/i}$ would ride in a limo to the dance.

(8) EACH$_{i}$ WRESTLER$_{i}$ TELL$_{j}$ SOMEONE$_{j}$ SWIMMER$_{j}$ THAT SOMEONE$_{k}$ FOOTBALL$_{k}$ PLAYER$_{k}$ ASK CAN IX$_{i}$ IX$_{j}$ IX$_{k}$ THREE-HUMANS-GO-TO DANCE (TOGETHER)
Each wrestler$_i$ told some swimmer$_j$ that some football player$_k$ asked if him$_i$ and him$_j$ and him$_k$ could go to the dance together.

Binding in ASL

- Every DP can be assigned a **locus** in space.
- Pronominal binding is realized by **pointing at the locus** which a DP has been assigned to (transcribed as IX).

A Counterexample in ASL? (Graf and Abner 2012)

Coordination of bound pronouns is grammatical in ASL.

(7) ALL$_i$ WRESTLER$_i$ INFORM$_j$ SOMEONE$_j$ SWIMMER$_j$ THAT IX$_{i/j}$ IX$_{j/i}$ WILL RIDE-IN-VEHICLE LIMO GO-TO DANCE
Every wrestler$_i$ told some swimmer$_j$ that him$_{i/j}$ and him$_{j/i}$ would ride in a limo to the dance.

(8) EACH$_i$ WRESTLER$_i$ TELL$_j$ SOMEONE$_j$ SWIMMER$_j$ THAT SOMEONE$_k$ FOOTBALL$_k$ PLAYER$_k$ ASK CAN IX$_i$ IX$_j$ IX$_k$ THREE-HUMANS-GO-TO DANCE (TOGETHER)
Each wrestler$_i$ told some swimmer$_j$ that some football player$_k$ asked if him$_i$ and him$_j$ and him$_k$ could go to the dance together.

Binding in ASL

- Every DP can be assigned a **locus** in space.
- Pronominal binding is realized by **pointing at the locus** which a DP has been assigned to (transcribed as IX).

The Role of Deixis

Pointing at referents in space resembles deictic pronouns in English. And deictic pronouns can easily be coordinated.

(9) Every/No/Some football player told every/some/no masseur that the coach wants to see him$_{deictic}$ and him$_{deictic}$ in his office.

Since **Limited Obviation** only applies to syntactic binding, (9) does not constitute a counterexample.

The Big Question

Are the coordinated pronouns in ASL **syntactically** bound?

The Role of Deixis

Pointing at referents in space resembles deictic pronouns in English. And deictic pronouns can easily be coordinated.

(9) Every/No/Some football player told every/some/no masseur that the coach wants to see him$_{deictic}$ and him$_{deictic}$ in his office.

Since **Limited Obviation** only applies to syntactic binding, (9) does not constitute a counterexample.

The Big Question

Are the coordinated pronouns in ASL **syntactically** bound?

Non-empty Domain Restrictions

While pronouns can be discourse-bound by quantifiers in English, the extension of the quantified DP must be non-empty.

(10) a. Every player is handed a card. He then has to role a dice.

b. # No player is handed a card. He then has to role a dice.

A similar pattern emerges for pronouns in ASL.

(11) EACH POLITICS PERSON$_i$ TELL-STORY (IX$_i$) WANT WIN
Each politician$_i$ said he$_i$ wants to win.

(12) NO POLITICS PERSON$_i$ TELL-STORY (?*IX$_i$) WANT WIN
No politician$_i$ said he$_i$ wants to win.

Further Suggestive Evidence

- only strict reading in ellipsis
- antecedent need not c-command

Caveats on Data

- "no DP" constructions generally disliked by most speakers
- ellipsis data only solid with non-quantified DPs
- lack of c-command data possibly due to telescoping (could not be tested with "no DP")

$\Rightarrow$ still work in progress

Further Suggestive Evidence

- only strict reading in ellipsis
- antecedent need not c-command

Caveats on Data

- "no DP" constructions generally disliked by most speakers
- ellipsis data only solid with non-quantified DPs
- lack of c-command data possibly due to telescoping (could not be tested with "no DP")

$\Rightarrow$ still work in progress

Section Summary

Are Merge-definable constraints sufficient?
Probably yes, if semantics isn't involved:

- Constraint results only hold for syntax
  $\Rightarrow$ Syntactic fragment of binding theory
- No discourse binding, no evaluation of specific readings
- Even then a limit on the number of required antecedents per binding domain is mandatory for Merge-definability $\Rightarrow$ **Limited Obviation**
- Satisfied in English and (probably) ASL
- A new descriptive universal for binding theory?

Outline

Reminder: Constraints through Operations

Constraints and operations are **closely connected**.

Theorem (Graf 2011; Kobele 2011)

A constraint C can be expressed via Merge iff C is MSO-definable.

- **Intuition**: Use feature calculus to emulate how information flows through the tree during computation
- Doable for almost all constraints from the syntactic literature
- Relies on symmetry of c-selection
  (category features & selection features)

head-argument relation $\equiv$ information pipeline

Reminder: Constraints through Operations

Constraints and operations are **closely connected**.

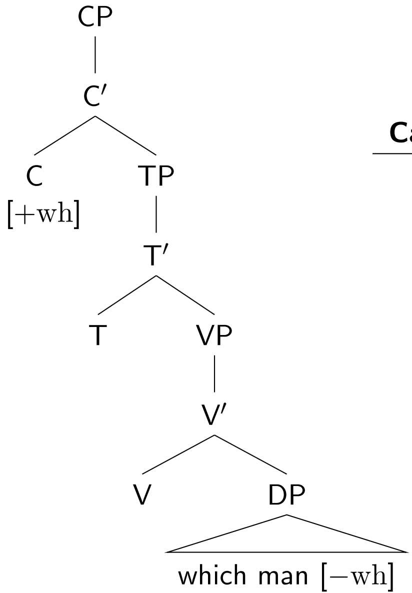
Theorem (Graf 2011; Kobele 2011)

A constraint C can be expressed via Merge iff C is MSO-definable.

- **Intuition**: Use feature calculus to emulate how information flows through the tree during computation
- Doable for almost all constraints from the syntactic literature
- Relies on symmetry of c-selection
(category features & selection features)

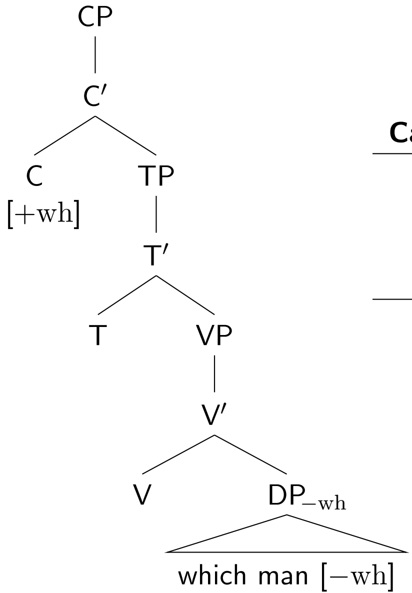
head-argument relation $\equiv$ information pipeline

Example: Keeping Track of Movers



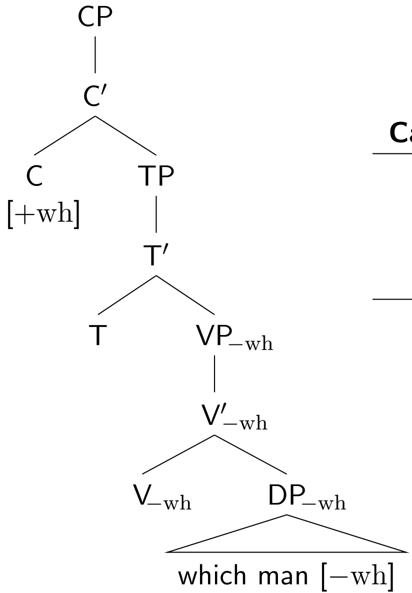
Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N

Example: Keeping Track of Movers



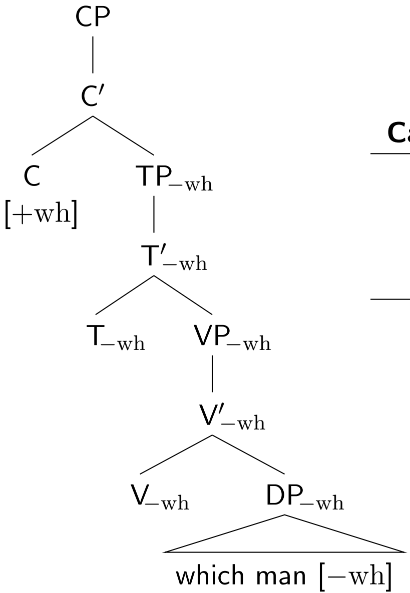
Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N
D$_{-\mathrm{wh}}$	N	

Example: Keeping Track of Movers



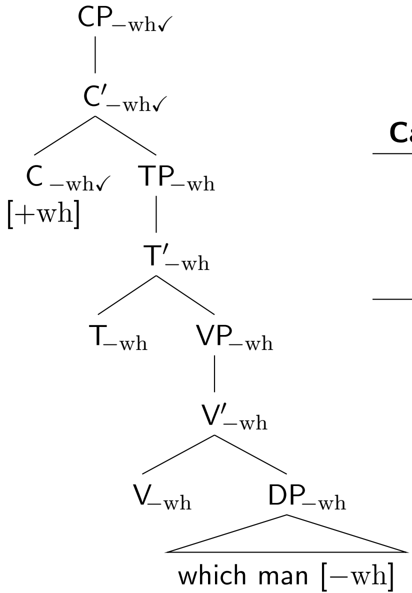
Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N
D$_{-\mathrm{wh}}$	N	V$_{-\mathrm{wh}}$
V$_{-\mathrm{wh}}$	D$_{-\mathrm{wh}}$	

Example: Keeping Track of Movers



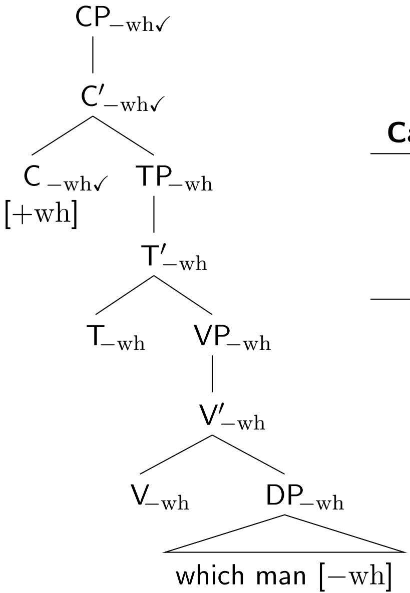
Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N
D$_{-\mathrm{wh}}$	N	V$_{-\mathrm{wh}}$
V$_{-\mathrm{wh}}$	D$_{-\mathrm{wh}}$	T$_{-\mathrm{wh}}$
T$_{-\mathrm{wh}}$	V$_{-\mathrm{wh}}$	

Example: Keeping Track of Movers



Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N
D_{-wh}	N	V_{-wh}
V_{-wh}	D_{-wh}	T_{-wh}
T_{-wh}	V_{-wh}	C_{-wh}
$C_{-wh✓}$	T_{-wh}	V,N

Example: Keeping Track of Movers



Category	Selects	Selected by
D	N	V
V	D	T
T	V	C
C	T	V,N
D_{-wh}	N	V_{-wh}
V_{-wh}	D_{-wh}	T_{-wh}
T_{-wh}	V_{-wh}	C_{-wh}
$C_{-wh\checkmark}$	T_{-wh}	V,N
C_{-wh}	T_{-wh}	V_{-wh},N_{-wh}

Adjuncts: The Price of Freedom

- Adjuncts very free: easily inserted, usually optional
- Freedom reflected in feature calculus, limits information flow
  $\Rightarrow$ feature calculus cannot emulate all constraints correctly

Semi-Permeability

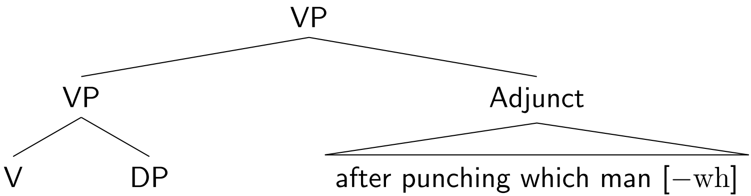
- Information flow into Adjuncts reliable
  $\Rightarrow$ Adjuncts can be selective about shape of tree
- Information flow out of Adjuncts unreliable
  $\Rightarrow$ Adjuncts cannot be depended on

Adjunct $\equiv$ black hole

Example: Adjunction a la Frey and Gärtner (2002)

Adjunction as Asymmetric Selection

Adjuncts select XP they adjoin to, but are not themselves selected.

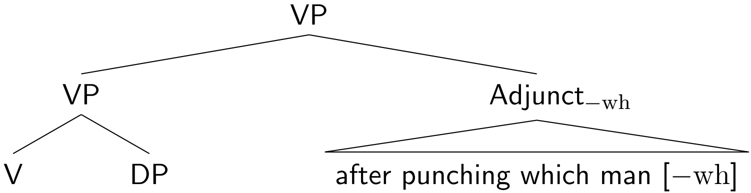


Category	Selects	Selected by
Adjunct	V	—
V	D	T

Example: Adjunction a la Frey and Gärtner (2002)

Adjunction as Asymmetric Selection

Adjuncts select XP they adjoin to, but are not themselves selected.

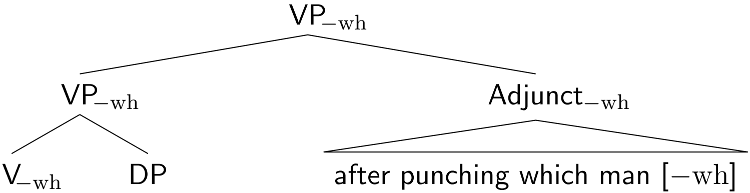


Category	**Selects**	**Selected by**
Adjunct	V	—
V	D	T
Adjunct$_{-\mathrm{wh}}$	V	—

Example: Adjunction a la Frey and Gärtner (2002)

Adjunction as Asymmetric Selection

Adjuncts select XP they adjoin to, but are not themselves selected.

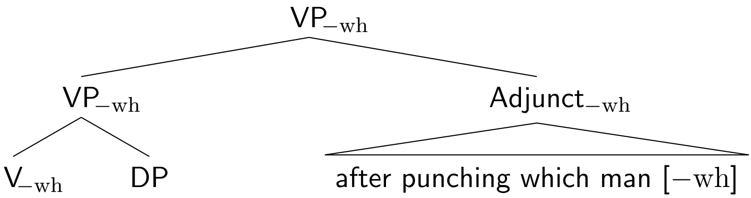


Category	Selects	Selected by
Adjunct	V	—
V	D	T
$\text{Adjunct}_{-\text{wh}}$	$\text{V}_{-\text{wh}}$	—
$\text{V}_{-\text{wh}}$	D	$\text{T}_{-\text{wh}}$

Example: Adjunction a la Frey and Gärtner (2002)

Adjunction as Asymmetric Selection

Adjuncts select XP they adjoin to, but are not themselves selected.



Category	**Selects**	**Selected by**
Adjunct	V	—
V	D	T
$\mathrm{Adjunct}_{-\mathrm{wh}}$	$\mathrm{V}_{-\mathrm{wh}}$	—
$\mathbf{V}_{-\mathrm{wh}}$	**D**	$\mathrm{T}_{-\mathrm{wh}}$

Defining Adjuncts

- **Optionality**
  Adjuncts can be omitted.

(13) **(Obviously)** I will **(easily)** ace this **((very) challenging)** exam **(because I (really) am that smart)**.

- **Independence**
  Independently well-formed adjuncts can be combined.

(14)
a. **Obviously** I will ace this exam.
b. I will **easily** ace this exam.
c. **Obviously** I will **easily** ace this exam.

Definition (Adjuncts)
Phrase marker *a* is an **Adjunct** iff it is optional and independent.

Defining Adjuncts

- **Optionality**
  Adjuncts can be omitted.

  (13) **(Obviously)** I will **(easily)** ace this **((very) challenging)** exam **(because I (really) am that smart)**.

- **Independence**
  Independently well-formed adjuncts can be combined.

  (14) a. **Obviously** I will ace this exam.
  b. I will **easily** ace this exam.
  c. **Obviously** I will **easily** ace this exam.

Definition (Adjuncts)

Phrase marker *a* is an **Adjunct** iff it is optional and independent.

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Adjunct Extension

What do these properties tell us about grammars with Adjuncts?
What is the general shape of the **generated language**?

Definition (Adjunct Extensions)

Let **s** and **t** be (multi-dominance) trees.
Then **t** is an **Adjunct extension** of **s** for grammar G (**s** $<_G$ **t**) iff **t** is the result of inserting one or more Adjuncts of G in **s**.

Example

- **Obviously** I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- I will ace this exam $<_G$ **Obviously** I will **easily** ace this exam
- **Obviously** I will ace this exam $\not<_G$ I will **easily** ace this exam
- I will ace this exam $\not<_G$ I will **easily** ace this test
- exam will this I ace $<_G$ **easily** exam will this I ace

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t***,*
then G generates **s***.*

Example

I will **easily** ace this **really** exam

I will **easily** ace this exam — I will ace this **really** exam

I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t**, *then G generates* **s**.

Example

I will **easily** ace this **really** exam

I will **easily** ace this exam

I will ace this **really** exam

I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t**, *then G generates* **s**.

Example

I will easily ace this really exam

✓ I will easily ace this exam

I will ace this really exam

I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t**, *then G generates* **s**.

Example

I will **easily** ace this **really** exam

✓ I will **easily** ace this exam

I will ace this **really** exam

I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for* G *and* G *generates* **t**, *then* G *generates* **s**.

Example

I will **easily** ace this **really** exam

✓ I will **easily** ace this exam

I will ace this **really** exam

✓ I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t***, then G generates* **s***.*

Example

I will **easily** ace this **really** exam

✓ I will **easily** ace this exam

* I will ace this **really** exam

✓ I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for* G *and* G *generates* **t***,*
then G *generates* **s***.*

Example

I will **easily** ace this **really** exam

✓ I will **easily** ace this exam

* I will ace this **really** exam

✓ I will ace this exam

Characterizing Adjunct Languages

Theorem (Optionality Closure)

If **t** *is an Adjunct extension of* **s** *for G and G generates* **t**, *then G generates* **s**.

Example

* I will **easily** ace this **really** exam

✓ I will **easily** ace this exam

* I will ace this **really** exam

✓ I will ace this exam

Characterizing Adjunct Languages [cont.]

Theorem (Independence Closure)

For **s** *and* **t** *adjunct extensions of some tree, G generates the "fusion" of* **s** *and* **t** *(*$\mathbf{s} \vee \mathbf{t}$*) if it generates both* **s** *and* **t***.*

Example

I **really** will **easily** ace this exam **now**

I will **easily** ace this exam

I **really** will ace this exam **now**

I will ace this exam

Characterizing Adjunct Languages [cont.]

Theorem (Independence Closure)

For **s** *and* **t** *adjunct extensions of some tree, G generates the "fusion" of* **s** *and* **t** *(*$\mathbf{s} \vee \mathbf{t}$*) if it generates both* **s** *and* **t**.

Example

I **really** will **easily** ace this exam **now**

✓ I will **easily** ace this exam

✓ I **really** will ace this exam **now**

✓ I will ace this exam

Characterizing Adjunct Languages [cont.]

Theorem (Independence Closure)

For **s** *and* **t** *adjunct extensions of some tree, G generates the "fusion" of* **s** *and* **t** *(*$\mathbf{s} \vee \mathbf{t}$*) if it generates both* **s** *and* **t**.

Example

I **really** will **easily** ace this exam **now**

✓ I will **easily** ace this exam

✓ I **really** will ace this exam **now**

✓ I will ace this exam

Characterizing Adjunct Languages [cont.]

Theorem (Independence Closure)

For **s** *and* **t** *adjunct extensions of some tree,* G *generates the "fusion" of* **s** *and* **t** *(*$\mathbf{s} \vee \mathbf{t}$*) if it generates both* **s** *and* **t***.*

Example

✓ I **really** will **easily** ace this exam **now**

✓ I will **easily** ace this exam

✓ I **really** will ace this exam **now**

✓ I will ace this exam

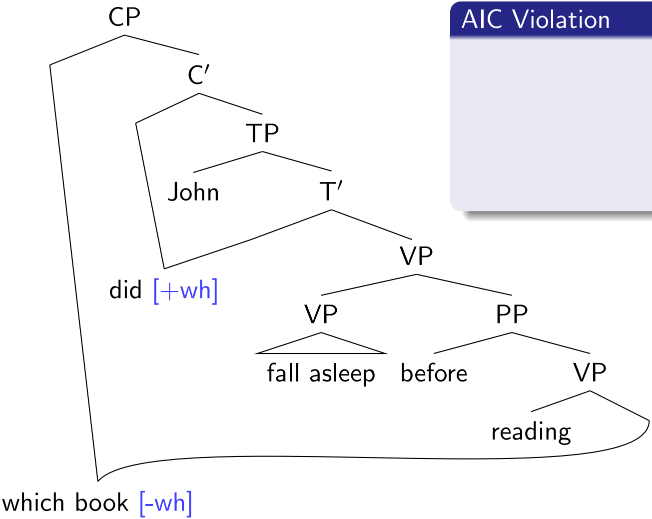
Interim Summary

Any implementation of Adjunction that captures Optionality and Independence yields a grammar formalism where

- $\Downarrow$ grammaticality is downward entailing with respect to $<_G$,
- $\Uparrow$ ungrammaticality is upward entailing with respect to $<_G$,
- $\vee$ grammaticality is preserved under "fusion".

Deriving the AIC

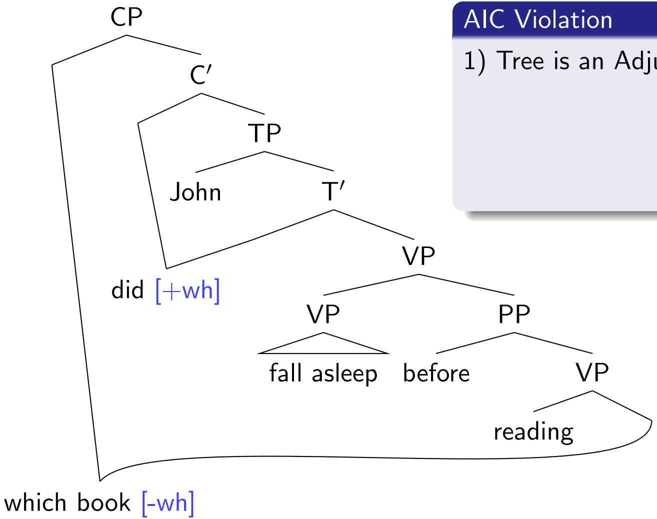
The AIC follows from **optionality closure and feature checking**.



AIC Violation

Deriving the AIC

The AIC follows from **optionality closure and feature checking**.

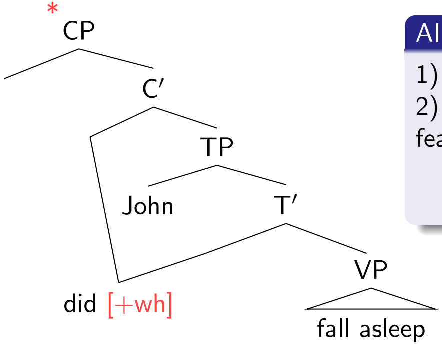


AIC Violation

1) Tree is an Adjunct extension

Deriving the AIC

The AIC follows from **optionality closure and feature checking**.

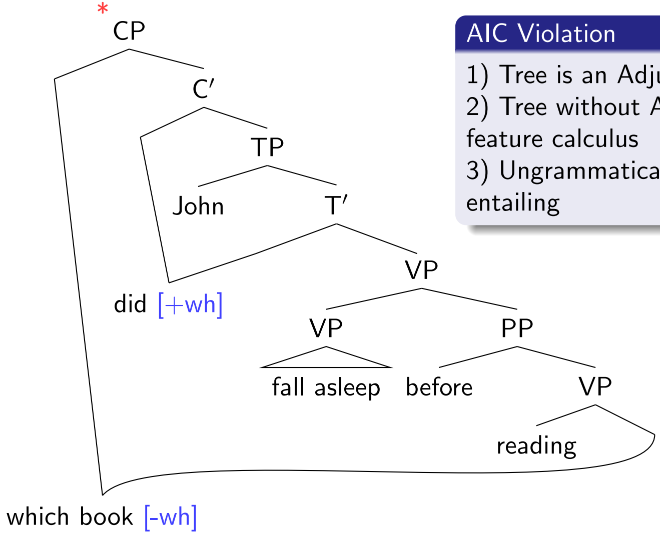


AIC Violation

1) Tree is an Adjunct extension
2) Tree without Adjunct violates feature calculus

Deriving the AIC

The AIC follows from **optionality closure and feature checking**.

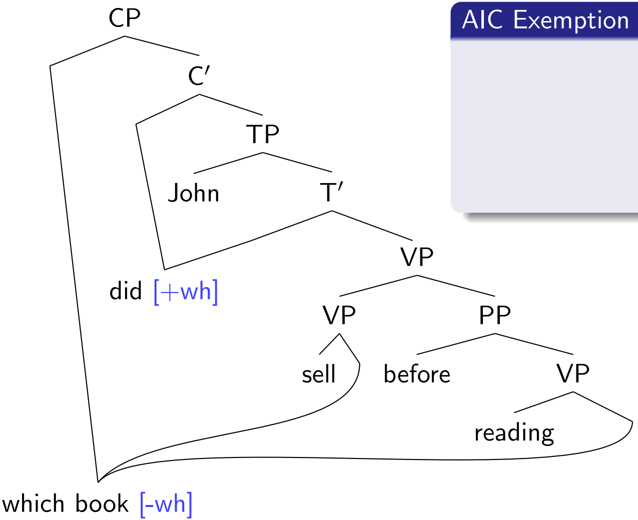


AIC Violation

1) Tree is an Adjunct extension
2) Tree without Adjunct violates feature calculus
3) Ungrammaticality is upward entailing

Why Parasitic Gaps are Different

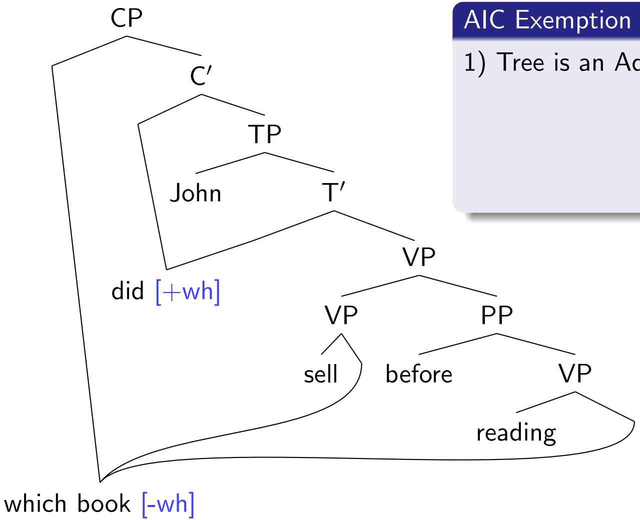
PGs piggyback on a **mandatory feature checker**.



AIC Exemption

Why Parasitic Gaps are Different

PGs piggyback on a **mandatory feature checker**.

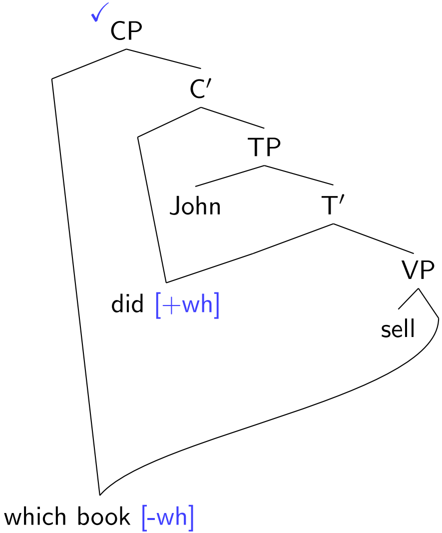


AIC Exemption

1) Tree is an Adjunct extension

Why Parasitic Gaps are Different

PGs piggyback on a **mandatory feature checker**.

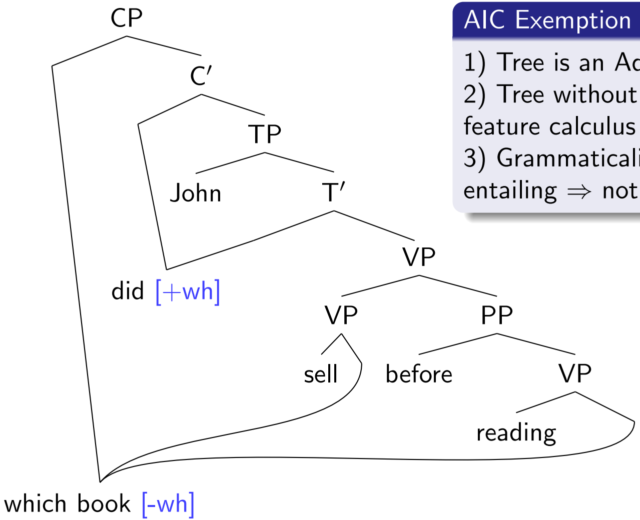


AIC Exemption

1) Tree is an Adjunct extension
2) Tree without Adjunct satisfies feature calculus

Why Parasitic Gaps are Different

PGs piggyback on a **mandatory feature checker**.



AIC Exemption

1) Tree is an Adjunct extension
2) Tree without Adjunct satisfies feature calculus
3) Grammaticality isn't upward entailing ⇒ nothing follows

Why Parasitic Gaps are Idempotent

Multiple PGs may piggyback on a single mover.

(15) Which movie did John **whilst mocking** throw in the trash **after watching**?

Follows from **independence closure**

(16) a. Which movie did John **whilst mocking** throw in the trash?

b. Which movie did John throw in the trash **after watching**?

No Obviation Between Pronouns in Different Adjuncts

Previous Observation on Binding

If sentence is good with 1 Adjunct containing a pronoun *p*, then it is good with *n* Adjuncts containing *p*.

(17)
a. No Egyptian goddess asked of some priest to sacrifice a goat for her.
b. No Egyptian goddess asked of some priest to sacrifice a goat in honor of her.
c. No Egyptian goddess asked of some priest to sacrifice a goat for her in honor of her.

Follows from independence closure, too

The Elephant in the Room

Not all adjuncts are Adjuncts
Some adjuncts can be extracted from (Truswell 2007):

(18) Which car did John drive Mary crazy **trying to fix**?

Truswell's event-based generalization ≈
some adjuncts more tightly integrated semantically

	sem-argument	sem-adjunct
syn-adjunct	Truswell adjuncts	Adjuncts
syn-argument	arguments	ECM adjuncts (?)

The Elephant in the Room

Not all adjuncts are Adjuncts

Some adjuncts can be extracted from (Truswell 2007):

(18) Which car did John drive Mary crazy **trying to fix**?

Truswell's event-based generalization ≈
some adjuncts more tightly integrated semantically

	sem-argument	sem-adjunct
syn-adjunct	Truswell adjuncts	Adjuncts
syn-argument	arguments	ECM adjuncts (?)

Another Problem: V2 in Germanic

(19) a. **Gestern** hat der Hans die Maria geküsst.
yesterday has the Hans the Maria kissed
'Yesterday, John kissed Mary.'

b. Hat der Hans die Maria geküsst?
has the Hans the Maria kissed
'Did John kiss Mary?'

c. * Hat der Hans die Maria geküsst.
has the Hans the Maria kissed
'John kissed Mary.'

Possible Answers

- V2 is post-syntactic and thus irrelevant for Optionality.
- V1 is grammatical, but restricted by discourse factors (e.g. telling jokes; works at least for Icelandic).

Another Problem: V2 in Germanic

(19) a. **Gestern** hat der Hans die Maria geküsst.
yesterday has the Hans the Maria kissed
'Yesterday, John kissed Mary.'

b. Hat der Hans die Maria geküsst?
has the Hans the Maria kissed
'Did John kiss Mary?'

c. * Hat der Hans die Maria geküsst.
has the Hans the Maria kissed
'John kissed Mary.'

Possible Answers

- V2 is post-syntactic and thus irrelevant for Optionality.
- V1 is grammatical, but restricted by discourse factors (e.g. telling jokes; works at least for Icelandic).

Relative Clauses

Relative clauses satisfy optionality and independence.

(20) a. the man
b. the man that I admire
c. the man that John works with
d. the man that John works with that I admire

As expected, they are islands.

(21) * Which politician do I dislike the reporter who interviewed.

Problems

There are exceptions, mostly based on information structure.

Coordinate Structure Constraint

Conjuncts are optional and independent *modulo agreement*.

(22) a. John and Bill and Mary left.

b. John and Bill left.

Suppose each *and XP* could have been inserted as an adjunct by the grammar. Then CSC $\equiv$ AIC and ATB $\equiv$ PG.

Example

Which beer did Ed brew and Greg drink

Which beer did Ed brew

Which beer did Greg drink

Problems

Agreement, NPI-licensing, binding dependencies, ...

$\Rightarrow$ conjuncts aren't always optional

Resumptive Pronouns

No island violations with resumptive pronoun instead of trace (e.g. Lebanese Arabic)

(23) ha-l-muttahame tfeeʒaʔto lamma/laʔanno
this-the-suspect.SgFem surprised.2 when/because
ʕrəfto ʔənno hiyye nhabasit.
know.2 that she imprisoned.3SgFem
'This suspect, you were surprised when/because you knew that she was imprisoned.' Aoun et al. (2001:575)

follows if binding rather than movement is involved

Problems

- Evidence for movement (weak crossover, parasitic gaps)
- Antecedent and adjunct must both be dropped for optionality to hold ⇒ discontinuous adjuncts (≈ TAG)?

Resumptive Pronouns

No island violations with resumptive pronoun instead of trace (e.g. Lebanese Arabic)

(23) ha-l-muttahame tfeeʒaʔto lamma/laʔanno
this-the-suspect.SgFem surprised.2 when/because
ʕrəfto ʔənno hiyye nhabasit.
know.2 that she imprisoned.3SgFem
'This suspect, you were surprised when/because you knew that she was imprisoned.' Aoun et al. (2001:575)

follows if binding rather than movement is involved

Problems

- Evidence for movement (weak crossover, parasitic gaps)
- Antecedent and adjunct must both be dropped for optionality to hold $\Rightarrow$ discontinuous adjuncts ($\approx$ TAG)?

Conclusion

- **Why do constraints exist at all?**
  Because constraints are a natural by-product of Merge.
- **Is Merge enough?**
  - For syntax: yes
  - For semantics/discourse: no
  - Concrete issue: Limited Obviation as a universal binding restriction
- **What else follows?**
  - If constraints are mediated by operations, the latter's properties affect the former
  - Concrete issue: Island constraints due to the limits of adjunction

References I

Aoun, Joseph, Lina Choueiri, and Norbert Hornstein. 2001. Resumption, movement and derivational economy. *Linguistic Inquiry* 32:371–403.

Conroy, Anastasia, Eri Takahashi, Jeffrey Lidz, and Colin Phillips. 2009. Equal treatment for all antecedents: How children succeed with Principle B. *Linguistc Inquiry* 40:446–486.

Doner, John. 1970. Tree acceptors and some of their applications. *Journal of Computer and System Sciences* 4:406–451.

Elbourne, Paul. 2005. On the acquisition of Principle B. *Linguistic Inquiry* 36:333–365.

Frey, Werner, and Hans-Martin Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of the Conference on Formal Grammar (FGTrento)*, 41–52. Trento.

Graf, Thomas. 2011. Closure properties of minimalist derivation tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 96–111. Heidelberg: Springer.

Graf, Thomas. 2012. Movement-generalized minimalist grammars. In *LACL 2012*, ed. Denis Béchet and Alexander J. Dikovsky, volume 7351 of *Lecture Notes in Computer Science*, 58–73.

References II

Graf, Thomas, and Natasha Abner. 2012. Is syntactic binding rational? In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 189–197.

Grodzinsky, Yosef, and Tanja Reinhart. 1993. The innateness of binding and coreference. *Linguistic Inquiry* 24:69–102.

Kobele, Gregory M. 2002. Formalizing mirror theory. *Grammars* 5:177–221.

Kobele, Gregory M. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 129–144.

Kobele, Gregory M. 2012. Idioms and extended transducers. In *Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, 153–161. Paris, France. URL `http://www.aclweb.org/anthology-new/W/W12/W12-4618`.

Kracht, Marcus. 1995. Is there a genuine modal perspective on feature structures? *Linguistics and Philosophy* 18:401–458.

Papafragou, Anna, and Julien Musolino. 2003. Scalar implicatures: Experiments at the semantics-pragmatics interface. *Cognition* 86:253–282.

References III

Papafragou, Anna, and Niki Tantalou. 2004. Children's computation of implicatures. *Language Acquisition* 12:71–82.

Potts, Christopher. 2001. Three kinds of transderivational constraints. In *Syntax at Santa Cruz*, ed. Séamas Mac Bhloscaidh, volume 3, 21–40. Santa Cruz: Linguistics Department, UC Santa Cruz.

Pullum, Geoffrey K. 2007. The evolution of model-theoretic frameworks in linguistics. In *Model-Theoretic Syntax @ 10*, ed. James Rogers and Stephan Kepser, 1–10.

Rogers, James. 1998. *A descriptive approach to language-theoretic complexity*. Stanford: CSLI.

Stabler, Edward P. 1997. Derivational minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer.

Stabler, Edward P. 2003. Comparing 3 perspectives on head movement. In *Syntax at sunset 3: Head movement and syntactic theory*, ed. A. Mahajan, volume 10 of *UCLA Working Papers in Linguistics*, 178–198. Los Angeles, CA: UCLA.

Stabler, Edward P. 2006. Sidewards without copying. In *Formal Grammar '06, Proceedings of the Conference*, ed. Gerald Penn, Giorgio Satta, and Shuly Wintner, 133–146. Stanford: CSLI Publications.

References IV

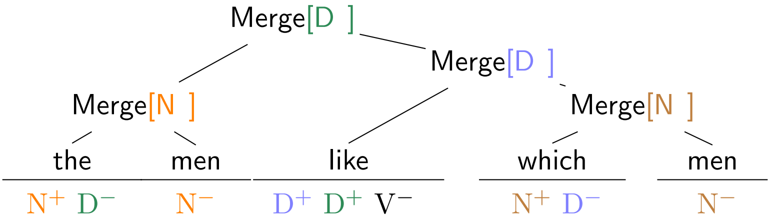
Stabler, Edward P. 2011. Computational perspectives on minimalism. In *Oxford handbook of linguistic minimalism*, ed. Cedric Boeckx, 617–643. Oxford: Oxford University Press.

Szendrői, Kriszta. 2004. Acquisition evidence for an interface theory of focus. In *Proceedings of GALA 2003*, ed. Jaqueline van Kampen and Sergio Bauuw, 457–468. Utrecht: LOT.

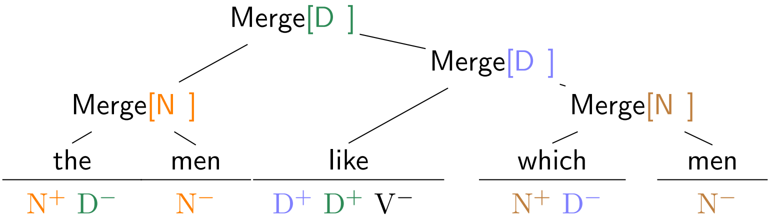
Thatcher, James W., and J. B. Wright. 1968. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory* 2:57–81.

Truswell, Robert. 2007. Tense, events, and extraction from adjuncts. In *Proceedings of the 43rd Annual Meeting of the Chicago Linguistic Society*.

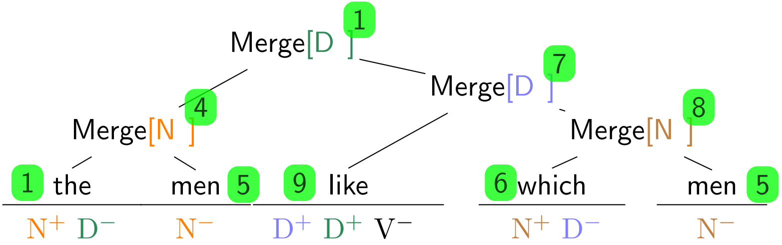
Compiling States into Features



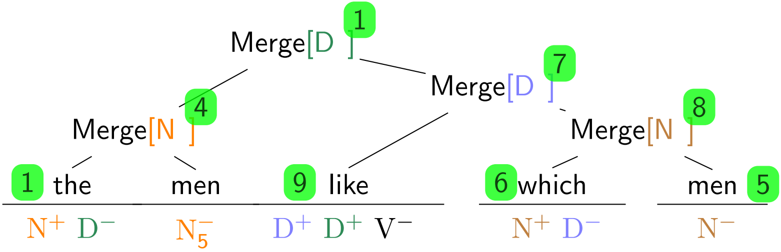
Compiling States into Features



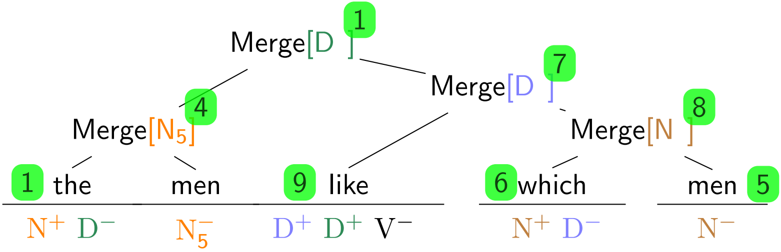
Compiling States into Features



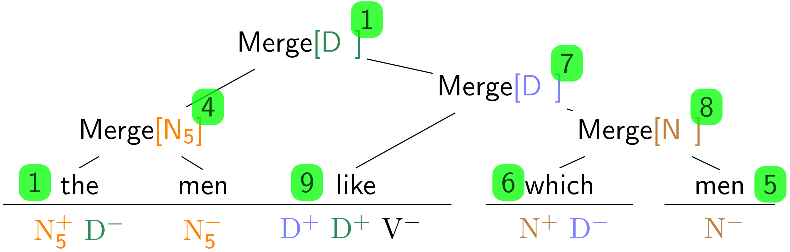
Compiling States into Features



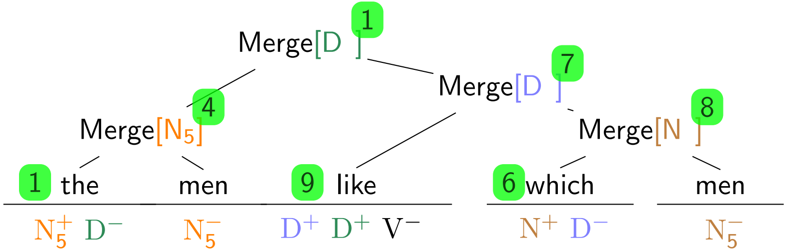
Compiling States into Features



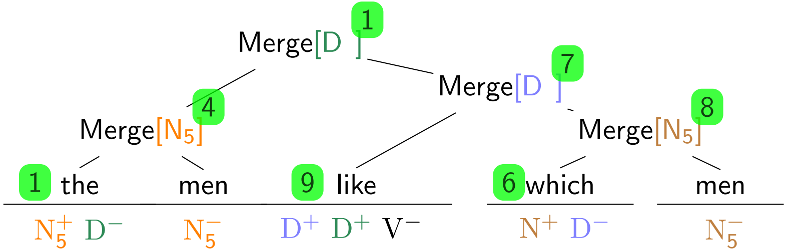
Compiling States into Features



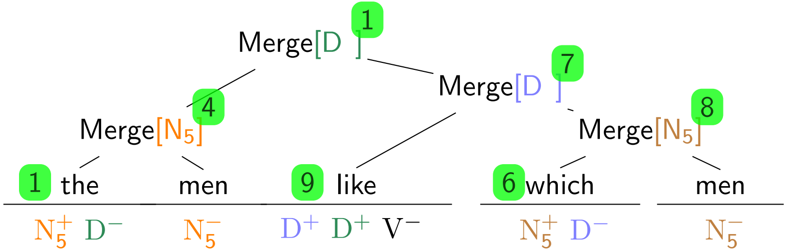
Compiling States into Features



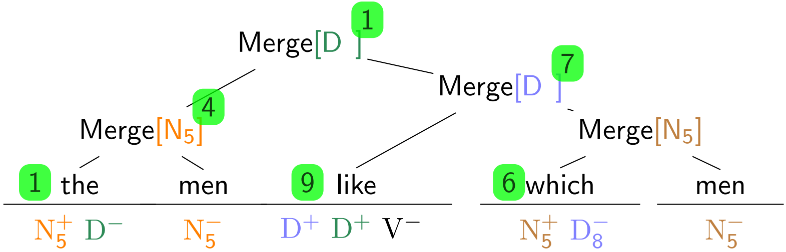
Compiling States into Features



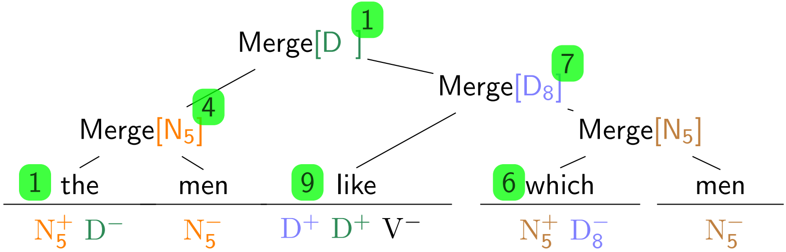
Compiling States into Features



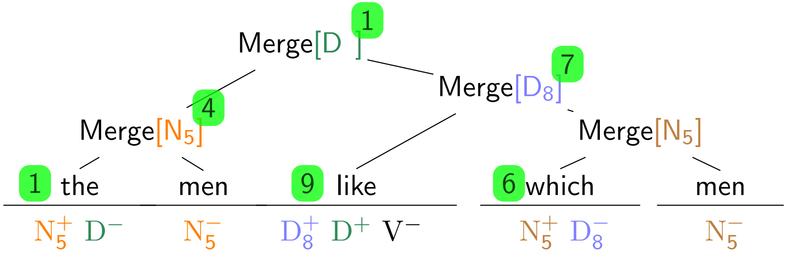
Compiling States into Features



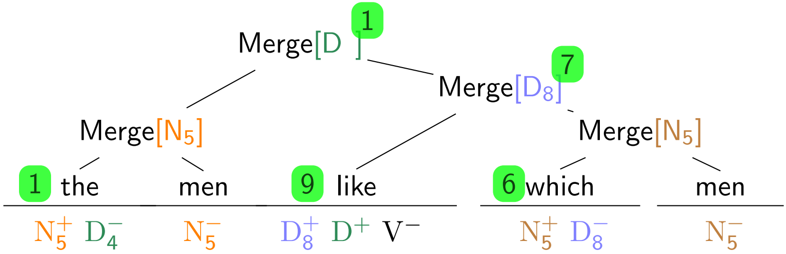
Compiling States into Features



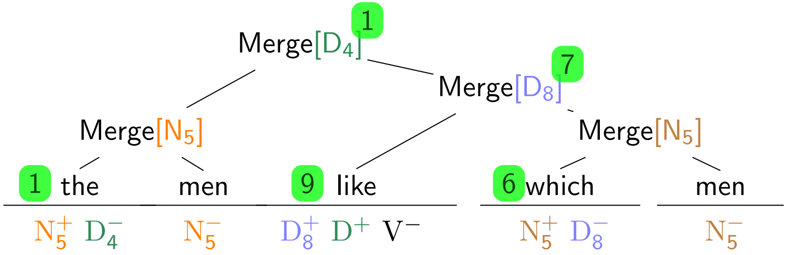
Compiling States into Features



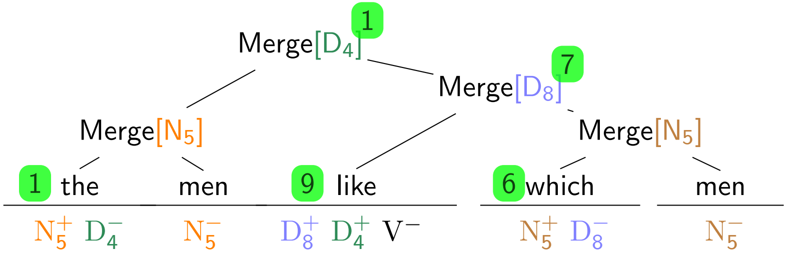
Compiling States into Features



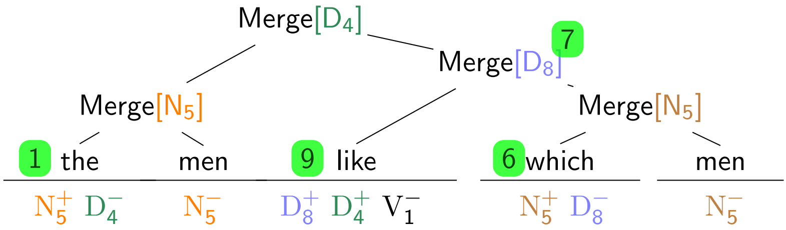
Compiling States into Features



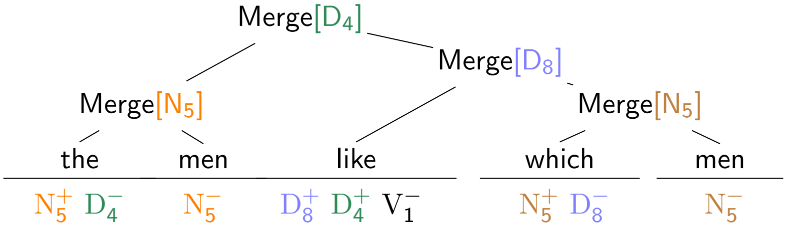
Compiling States into Features



Compiling States into Features



Compiling States into Features



Representational $\equiv$ Derivational

MSO over Representations	**MSO over Derivations**
$\wedge, \vee, \neg, \rightarrow$	$\wedge, \vee, \neg, \rightarrow$
$\exists, \forall$	$\exists, \forall$
$\equiv$	$\equiv$
standard dominance $\triangleleft$	derivational dominance $\blacktriangleleft$

- $x \triangleleft y$ iff $\phi(x, y)$, where ϕ uses $\blacktriangleleft$ but not $\triangleleft$
  $\Rightarrow$ replacing each occurrence of $x \triangleleft y$ by $\phi(x, y)$
  in representational constraint C yields derivational C'
- $x \blacktriangleleft y$ iff $\psi(x, y)$, where ψ uses $\triangleleft$ but not $\blacktriangleleft$
  $\Rightarrow$ replacing each occurrence of $x \blacktriangleleft y$ by $\psi(x, y)$
  in derivational constraint C yields representational C'

Derivational $\equiv$ Transderivational

A Different Perspective on Transderivationality

Transderivational constraints do not filter out suboptimal trees. They **rewrite suboptimal trees into optimal ones**.

- Rewrite procedure carried out by **linear tree transducer**
- Given a set of Minimalist derivations as inputs, transducer produces set of outputs that can be computed by a tree automaton
- Compile said automaton into the features as usual

(Dis)Advantages of Constraint Classes

Are constraints redundant? Should we just use feature checking?

- **Shortcomings of Local Constraints**
  less succinct, often incomprehensible, hide generalizations
- **Shortcomings of Derivational Constraints**
  some constraints are significantly more complicated when stated over derivations (e.g. ECP)
- **Advantages of Transderivational Constraints**
  can state generalizations **across grammars** that are not expressible with derivational/representational constraints

Methodological Moral of the Story

Even though the constraint classes have the same power, they each have their own advantages and disadvantages.
⇒ Use the type of constraint that is best suited to the task!

A Purely Transderivational Generalization

Shortest Derivation Principle

Given a set of competing derivations, pick the one with the fewest instances of Move.

Toy Grammar 1

- At least one DP moves out of VP.
- Two options:
  - Move to SpecYP, and YP then moves to SpecZP (roll-up)
  - Move directly to SpecZP (one-fell-swoop)
- Result: Exactly one DP must move from VP to SpecZP.

Toy Grammar 2

- At most one DP moves out of VP, directly into SpecZP.
- Result: No DP may move from VP to SpecZP.

A Purely Transderivational Generalization

Shortest Derivation Principle

Given a set of competing derivations, pick the one with the fewest instances of Move.

Toy Grammar 1

- At least one DP moves out of VP.
- Two options:
  - Move to SpecYP, and YP then moves to SpecZP (roll-up)
  - Move directly to SpecZP (one-fell-swoop)
- Result: Exactly one DP must move from VP to SpecZP.

Toy Grammar 2

- At most one DP moves out of VP, directly into SpecZP.
- Result: No DP may move from VP to SpecZP.

A Purely Transderivational Generalization

Shortest Derivation Principle

Given a set of competing derivations, pick the one with the fewest instances of Move.

Toy Grammar 1

- At least one DP moves out of VP.
- Two options:
  - Move to SpecYP, and YP then moves to SpecZP (roll-up)
  - Move directly to SpecZP (one-fell-swoop)
- Result: Exactly one DP must move from VP to SpecZP.

Toy Grammar 2

- At most one DP moves out of VP, directly into SpecZP.
- Result: No DP may move from VP to SpecZP.

Implications for Acquisition

- "cognitive load explanations" for acquisition delays of Principle B and Focus Projection
  (Grodzinsky and Reinhart 1993; Szendrői 2004)
  - Certain processes involve transderivational constraints.
  - Comparing multiple trees too computationally demanding for young children
    $\Rightarrow$ delay in acquisition due to insufficient working memory
- Implausible explanation if transderivational $\equiv$ derivational (no extra processing load)
- Recent findings: Principle B delay an artifact of experimental setup
  (Papafragou and Musolino 2003; Papafragou and Tantalou 2004; Elbourne 2005; Conroy et al. 2009)
- Same problem with Focus experiment?
  A pilot study is in preparation.

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \lor \text{herself}(x) \lor \text{itself}(x)$$

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \land \neg\,(x \lhd y) \land \forall z\, [z \lhd x \rightarrow z \lhd y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

$$\text{c-com}(x, y) \leftrightarrow \neg\, (x \approx y) \wedge \neg\, (x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor

$$\text{c-com}(x, y) \leftrightarrow \neg (x \approx y) \wedge \neg (x \triangleleft y) \wedge \forall z \, [z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \lor \text{herself}(x) \lor \text{itself}(x)$$

"x is an anaphor iff

$$\text{c-com}(x, y) \leftrightarrow \neg (x \approx y) \land \neg (x \triangleleft y) \land \forall z \, [z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*

or

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself*

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \lhd y) \wedge \forall z\, [z \lhd x \rightarrow z \lhd y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or

$$\text{c-com}(x, y) \leftrightarrow \neg (x \approx y) \wedge \neg (x \triangleleft y) \wedge \forall z \, [z \triangleleft x \rightarrow z \triangleleft y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \lhd y) \wedge \forall z\,[z \lhd x \rightarrow z \lhd y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \lor \text{herself}(x) \lor \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \land \neg\,(x \lhd y) \land \forall z\,[z \lhd x \rightarrow z \lhd y]$$

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg (x \approx y) \wedge \neg (x \triangleleft y) \wedge \forall z \, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \lhd y) \wedge \forall z\, [z \lhd x \rightarrow z \lhd y]$$

"x c-commands y iff
it is not the case that

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\, (x \approx y) \wedge \neg\, (x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff

it is not the case that x and y are the same node,

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\, (x \approx y) \wedge \neg\, (x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,
and

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,
and every z

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \lor \text{herself}(x) \lor \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg(x \approx y) \land \neg(x \triangleleft y) \land \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,
and every z that dominates x

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself*
or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\, [z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,
and every z that dominates x also

Example: Stating Principle A

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\text{anaphor}(x) \leftrightarrow \text{himself}(x) \vee \text{herself}(x) \vee \text{itself}(x)$$

"x is an anaphor iff x is labeled *himself* or x is labeled *herself* or x is labeled *itself*."

$$\text{c-com}(x, y) \leftrightarrow \neg\,(x \approx y) \wedge \neg\,(x \triangleleft y) \wedge \forall z\,[z \triangleleft x \rightarrow z \triangleleft y]$$

"x c-commands y iff
it is not the case that x and y are the same node,
and it is not the case that x dominates y,
and every z that dominates x also dominates y."

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg \exists Z [y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \;\leftrightarrow\; \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg \exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y
and

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that contains y

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that contains y and

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\text{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \text{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \text{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that contains y and is a TP

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that contains y and is a TP and

Set Quantification: Talking About Domains

Problem: domains are **sets of nodes** $\Rightarrow$ move beyond FO

Monadic Second-Order Logic for Trees (MSO)

Extension of FO with

$X, Y, Z, \ldots$	set variables $X, Y, Z, \ldots$
$\mathrm{YP}(X)$	set X is a YP
$\exists, \forall$	there is a set, for all sets
$\in, \subset$	set containment, proper subset

$$\text{b-dom}(X, y) \leftrightarrow \mathrm{TP}(X) \wedge y \in X \wedge \neg\exists Z[y \in Z \wedge \mathrm{TP}(Z) \wedge Z \subset X]$$

"X is the binding domain of y iff X is a TP and X contains y and there is no Z that contains y and is a TP and is a proper subset of X."

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z]\big]\Big]$$

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z]\big]\Big]$$

“For every x that is an anaphor

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that

there is a y

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP,

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \\ \exists Z[\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z [\text{b-dom}(Z, x) \wedge y \in Z] \big] \Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z]\big]\Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x and

Principle A Again

Principle A (slightly simplified)

Every anaphor must be c-commanded by some DP within its binding domain.

$$\forall x \Big[\text{anaphor}(x) \rightarrow \exists y \big[\text{c-com}(y, x) \wedge \text{DP}(y) \wedge \exists Z[\text{b-dom}(Z, x) \wedge y \in Z]\big]\Big]$$

"For every x that is an anaphor it holds that
there is a y that c-commands x and is labeled DP, and
there is a Z that is the binding domain of x and contains y."

Principle A is Easy

- already saw how Principle A can be expressed in MSO
  $\Rightarrow$ SE/SELF anaphors no problem
- What about long distance reflexives?
  - Icelandic type: usually allows local binding $\Rightarrow$ like SE/SELF

    (24) Jón$_i$ segir að María$_j$ elski sig$_{i/j}$.
    Jon says that Maria loves.SUBJ SE
    'Jon says that Maria loves him/herself.'

  - Swedish type: no local binding allowed $\Rightarrow$ like pronouns

    (25) Generalen$_i$ tvingade översten$_j$ PRO$_j$ att hjälpa sig$_{i/*j}$.
    General.the forced colonel.the PRO to help SE
    'The general forced the colonel to help him(*self).'

Two Common Questions on ASL Binding

What about other obviation domains in ASL?

Coordination and nested VP/TP domains provide the only true ASL parallel to their English counterparts, the latter of which also introduce new binding domains in ASL. Nested DP structures are not well-attested in the language and comparable adjunct structures are expressed in ASL through the use of complex locative and classifier morphology.

Could spatial reference just be an elaborate case or gender system?

Then the grammatical coordination examples parallel the coordination of *him* and *her* in English and are not a problem for Limited Obviation. However, there is no sense in which spatial loci are inherently associated with (pro)nominals in ASL, as is typical of gender systems, nor are spatial loci reliably assigned in specific syntactic environments, as is typical of case systems.

More on Coordination in English

Some speakers accept (26) as grammatical.

(26) ?? Every/No/Some football player told every/no/some masseur that the coach wants him to run six laps and him to prepare the massage room.

If this pattern is a productive instance of coordinating syntactically bound pronouns, it would falsify **Limited Obviation**.
But just like in ASL, the binding mechanism at play here arguably isn't (purely) syntactic in nature.

- Most speakers need to put (contrastive) stress on the respective pronouns.
- *No* seems to be disprefered compared to *every* and *some*.
- There is no c-command requirement (even in configurations where QR is bounded).

More on Coordination in English [cont.]

(27)

a. A coach of every/some football player told a receptionist of every/some masseur that the team's president wants him to get a massage and him to give it.

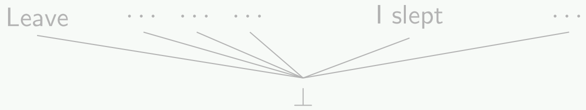
b. An agent of every/some actress told a bodyguard of every/some first lady that he wants her to do a movie about Jackie Kennedy and her to be on the set as a consultant.

c. An interview that every/some football player liked included a quib, which every/some masseur had related to the reporter at some point, that the coach always ordered him to run six laps and him to prepare the massage chair.

Adjunct Algebra Induced by G

- Order the set of all possible (not necessarily grammatical) trees by G's Adjunct extension relation.
- Add a dummy element $\perp$ at the bottom.

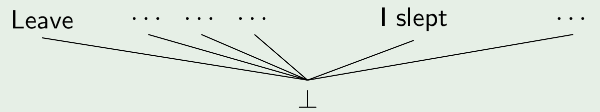
Example



Adjunct Algebra Induced by G

- Order the set of all possible (not necessarily grammatical) trees by G's Adjunct extension relation.
- Add a dummy element $\perp$ at the bottom.

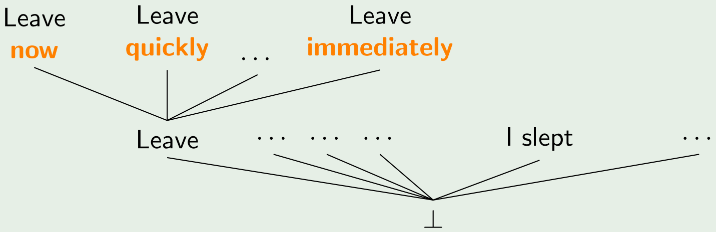
Example



Adjunct Algebra Induced by G

- Order the set of all possible (not necessarily grammatical) trees by G's Adjunct extension relation.
- Add a dummy element $\perp$ at the bottom.

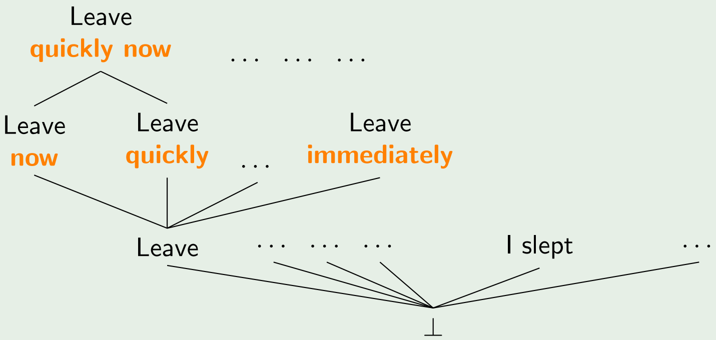
Example



Adjunct Algebra Induced by G

- Order the set of all possible (not necessarily grammatical) trees by G's Adjunct extension relation.
- Add a dummy element $\perp$ at the bottom.

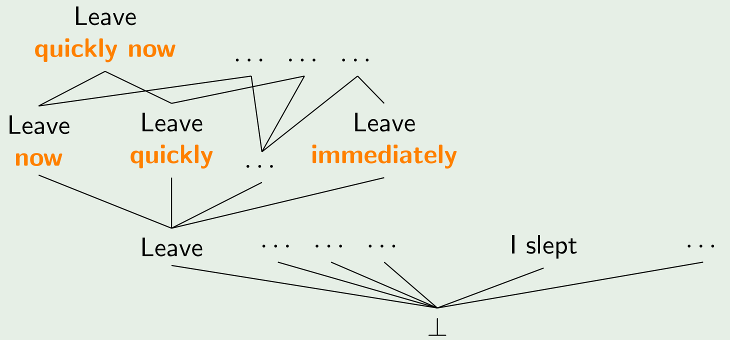
Example



Adjunct Algebra Induced by G

- Order the set of all possible (not necessarily grammatical) trees by G's Adjunct extension relation.
- Add a dummy element $\perp$ at the bottom.

Example



Adjunct Languages are Collections of Ideals

Definition (Ideal)

A non-empty subset S of a poset $\langle A, \leq \rangle$ is an **ideal** iff

- for every $x \in S$, $y \leq x$ implies $y \in S$, and
- for all $x, y \in S$ there is some $z \in S$ such that $x \leq z$ and $y \leq z$.

Theorem

The tree language generated by grammar G is a collection of ideals over the Adjunct Algebra induced by G (modulo $\perp$).

Parallels to Logical And

- **Grammaticality is Downward Entailing**
  $a \wedge b = 1$ implies $a = 1$
- **Ungrammaticality is Upward Entailing**
  $a = 0$ implies $a \wedge b = 0$
- **Grammaticality is Preserved Under "Fusion"**
  $a \wedge b = 1$ and $a \wedge c = 1$ jointly imply $a \wedge b \wedge c = 1$