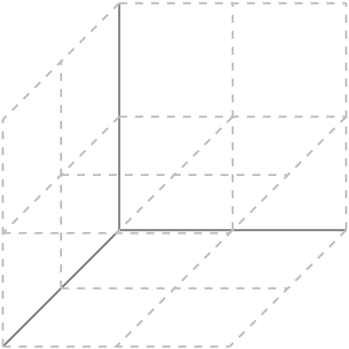
Some Interdefinability Results for Syntactic Constraint Classes

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

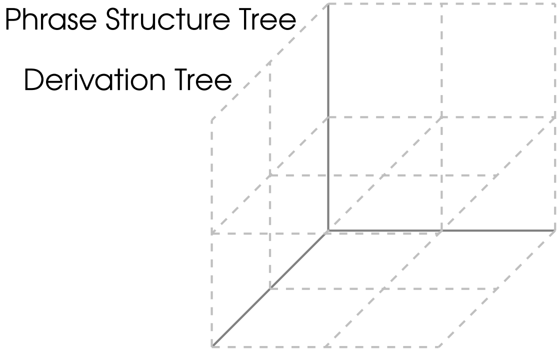
University of California, Los Angeles

Mathematics of Language 11
Bielefeld, Germany

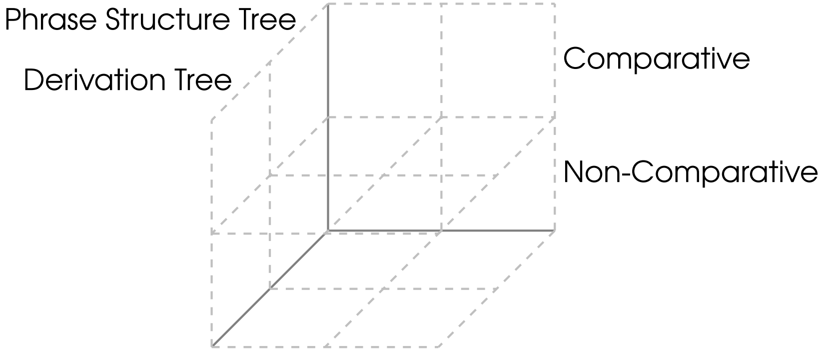
Constraint Classes (Müller and Sternefeld 2000)



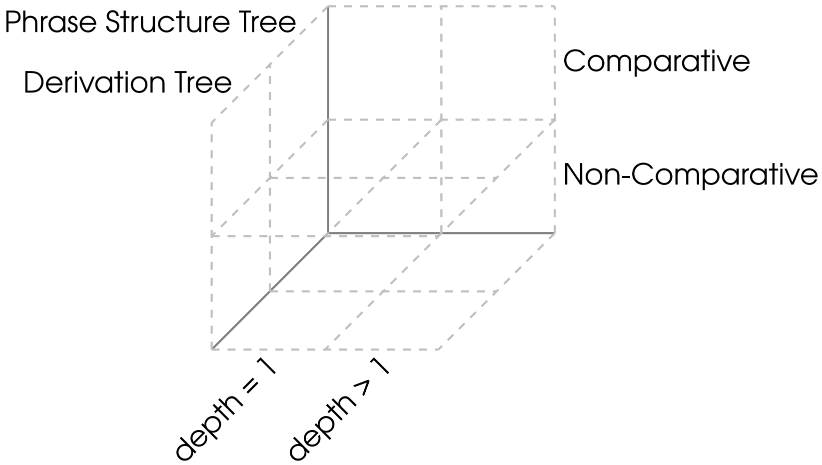
Constraint Classes (Müller and Sternefeld 2000)



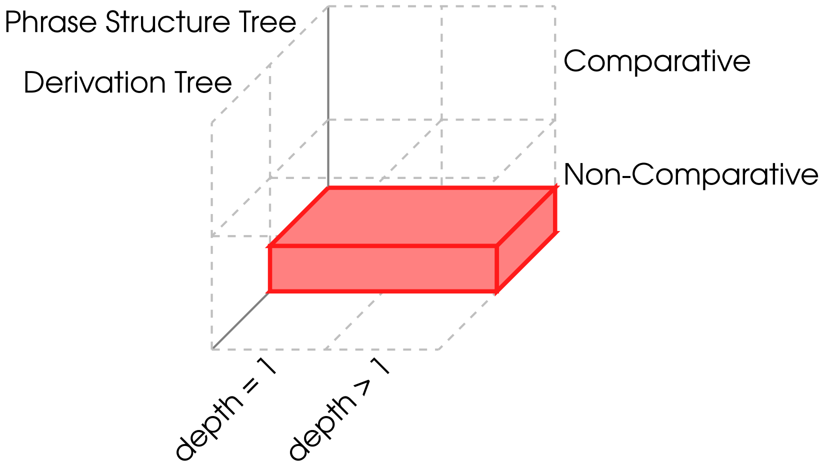
Constraint Classes (Müller and Sternefeld 2000)



Constraint Classes (Müller and Sternefeld 2000)

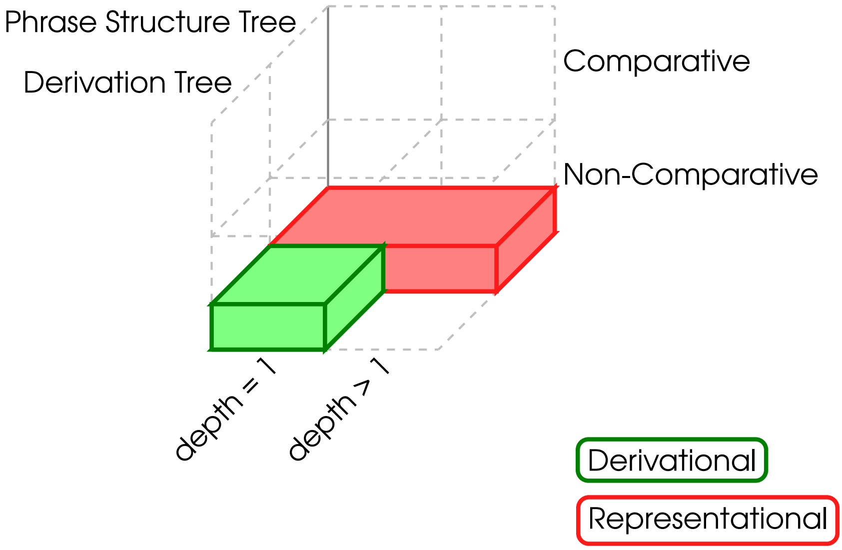


Constraint Classes (Müller and Sternefeld 2000)

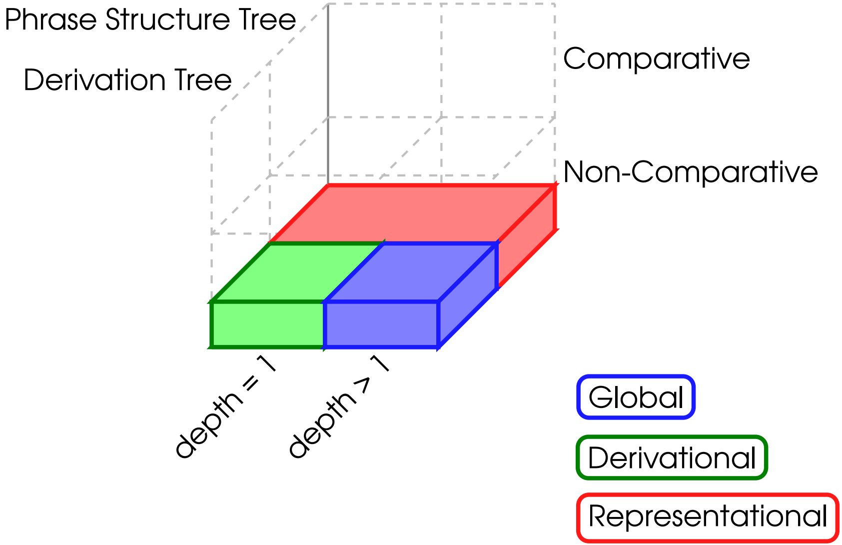


Representational

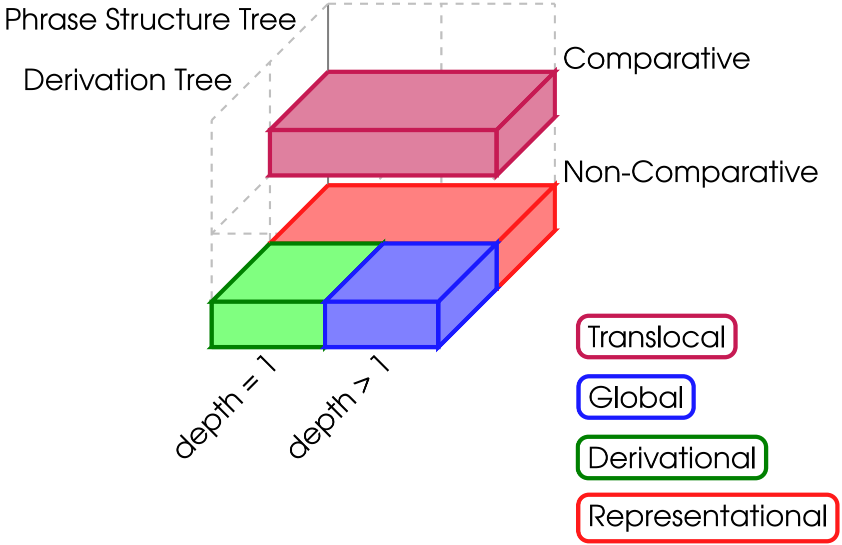
Constraint Classes (Müller and Sternefeld 2000)



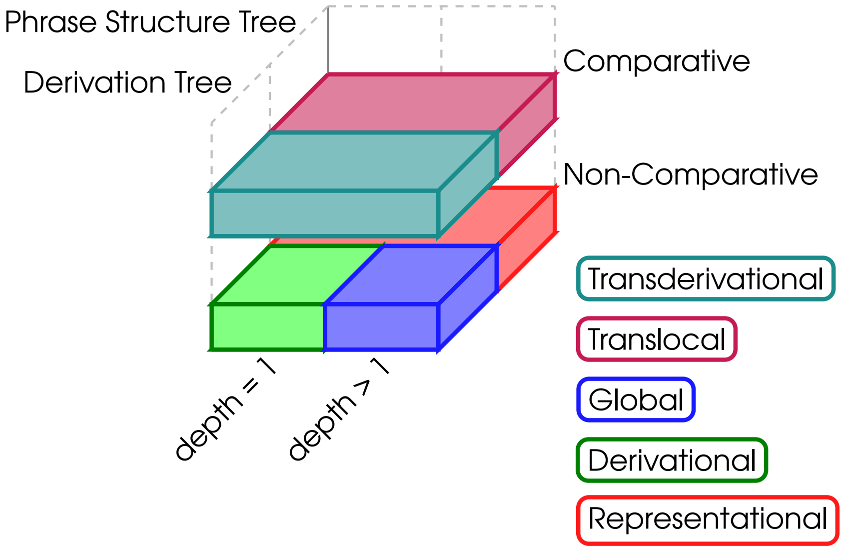
Constraint Classes (Müller and Sternefeld 2000)



Constraint Classes (Müller and Sternefeld 2000)



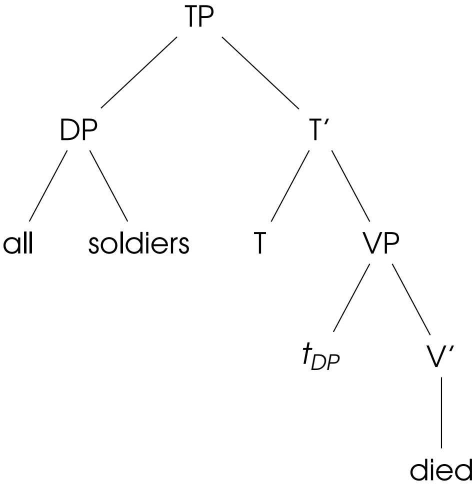
Constraint Classes (Müller and Sternefeld 2000)



Constraints on Phrase Structure Trees

Representational constraints

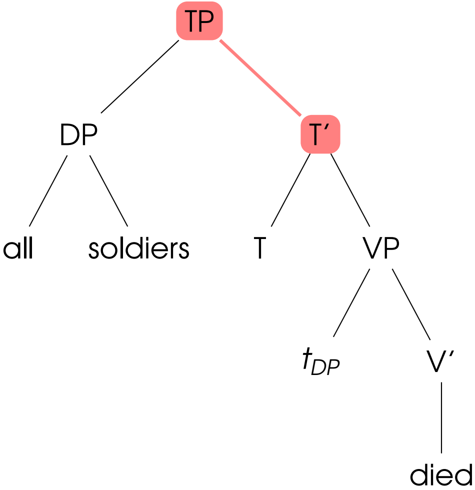
- Projection constraints
- Selection constraints
- Proper Government
- Island conditions
- . . .



Constraints on Phrase Structure Trees

Representational constraints

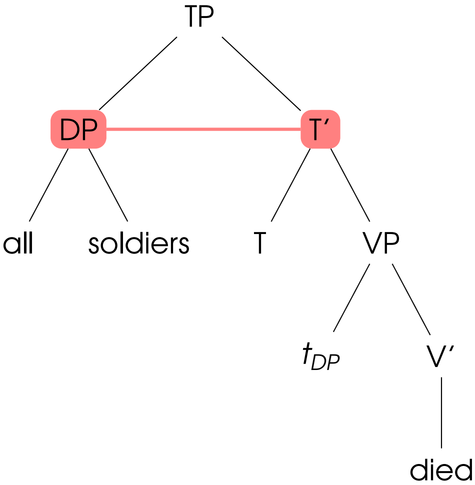
- Projection constraints
- Selection constraints
- Proper Government
- Island conditions
- ...



Constraints on Phrase Structure Trees

Representational constraints

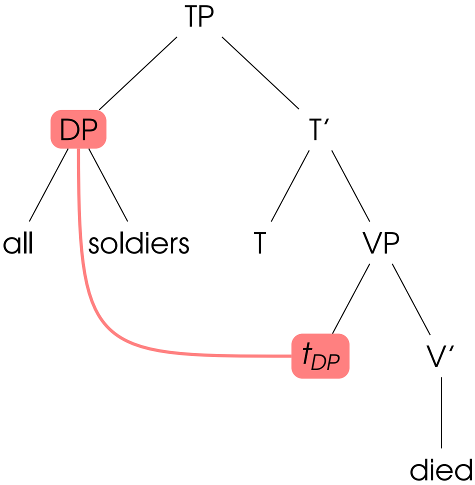
- Projection constraints
- Selection constraints
- Proper Government
- Island conditions
- ...



Constraints on Phrase Structure Trees

Representational constraints

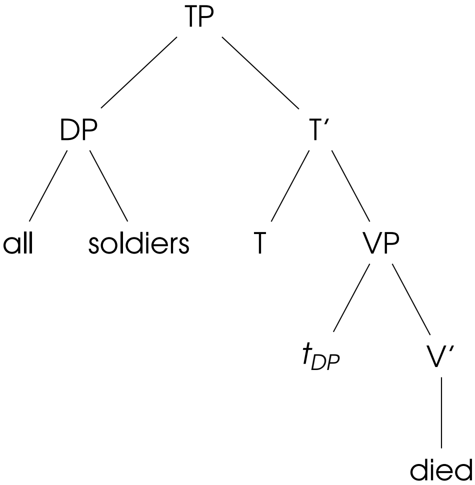
- Projection constraints
- Selection constraints
- Proper Government
- Island conditions
- ...



Constraints on Phrase Structure Trees

Representational constraints

- Projection constraints
- Selection constraints
- Proper Government
- Island conditions
- . . .



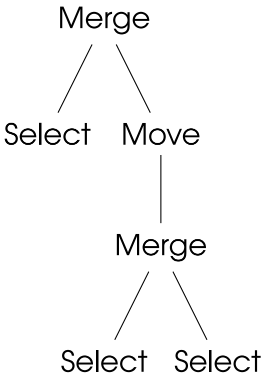
Constraints on Derivation Trees

Derivational constraints

- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints

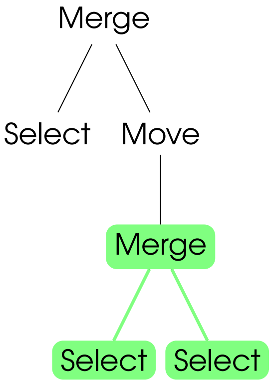
- Look-ahead constraints
- Projection Principle
- ...



Constraints on Derivation Trees

Derivational constraints
- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints
- Look-ahead constraints
- Projection Principle
- ...



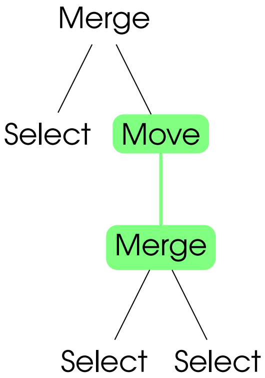
Constraints on Derivation Trees

Derivational constraints

- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints

- Look-ahead constraints
- Projection Principle
- ...



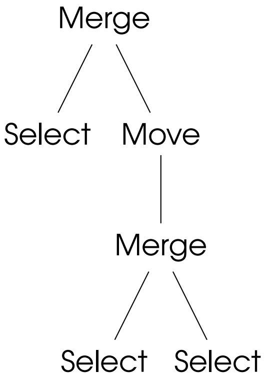
Constraints on Derivation Trees

Derivational constraints

- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints

- Look-ahead constraints
- Projection Principle
- ...



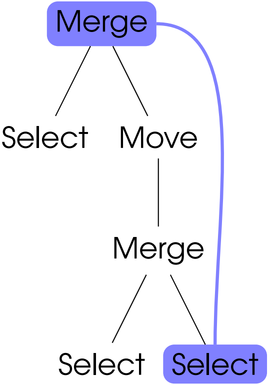
Constraints on Derivation Trees

Derivational constraints

- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints

- Look-ahead constraints
- Projection Principle
- ...



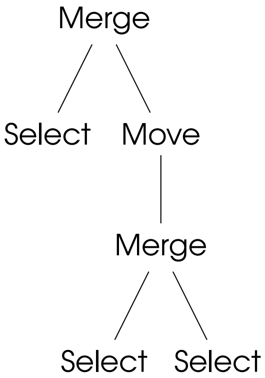
Constraints on Derivation Trees

Derivational constraints

- Definition of Merge
- Shortest Move
- Specifier Island Constraint
- A-over-A Constraint
- ...

Global constraints

- Look-ahead constraints
- Projection Principle
- ...

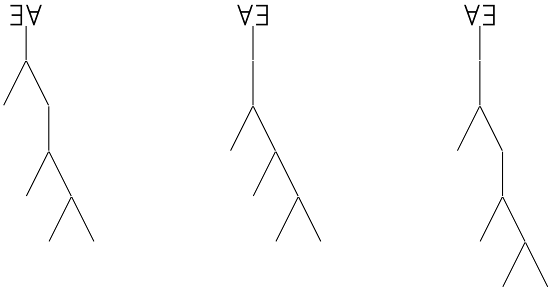


Comparative Constraints

Fewest Steps

Given two well-formed derivations that were created from the same argument structure and have the same PF- and LF-yield, pick the one with fewer instances of Move.

Every chicken has a head.

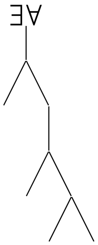


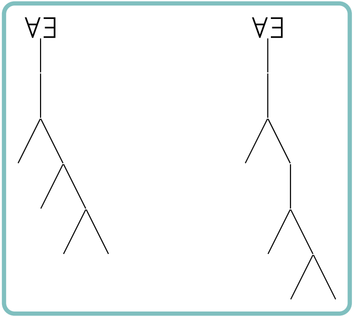
Comparative Constraints

Fewest Steps

Given two well-formed derivations that were created from the same argument structure and have the same PF- and LF-yield, pick the one with fewer instances of Move.

Every chicken has a head.



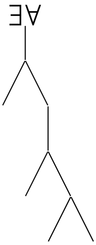


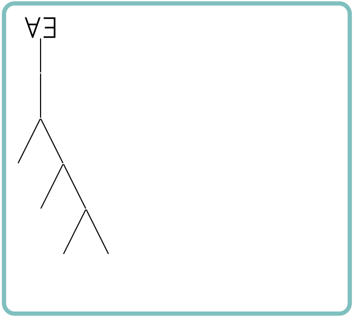
Comparative Constraints

Fewest Steps

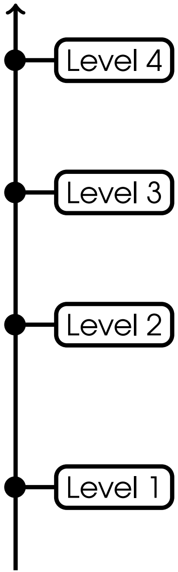
Given two well-formed derivations that were created from the same argument structure and have the same PF- and LF-yield, pick the one with fewer instances of Move.

Every chicken has a head.

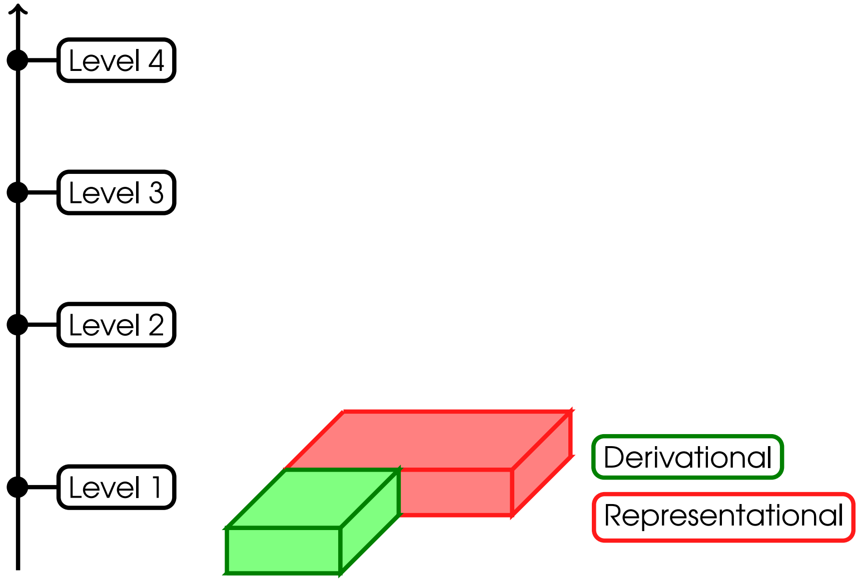




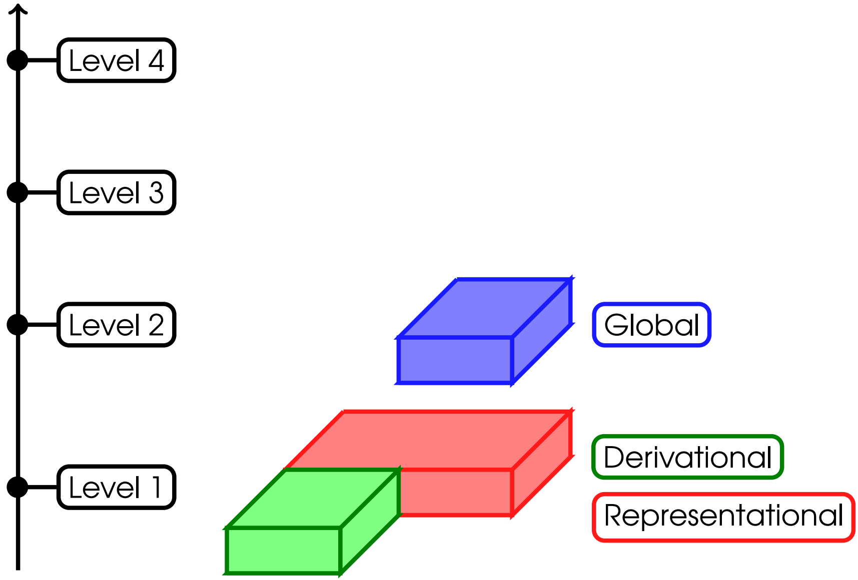
The Müller-Sternefeld-Hierarchy (Müller 2005)



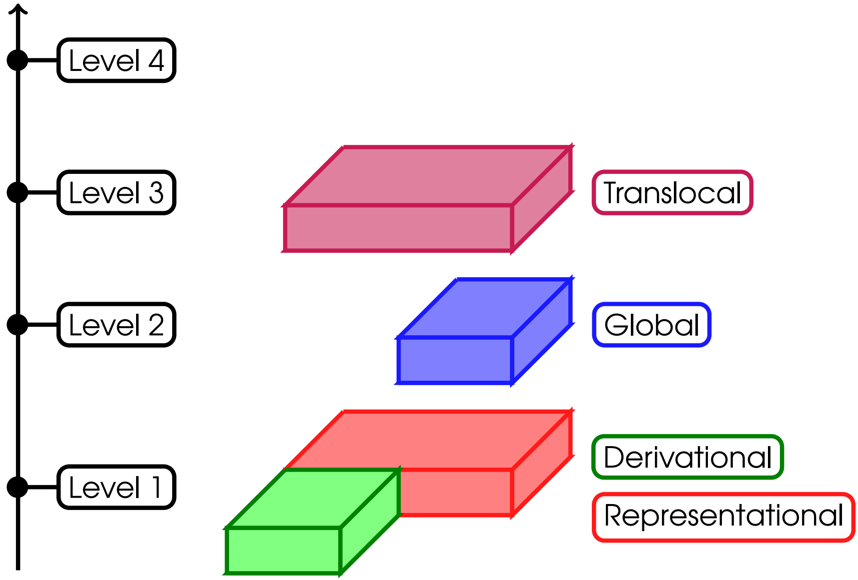
The Müller-Sternefeld-Hierarchy (Müller 2005)



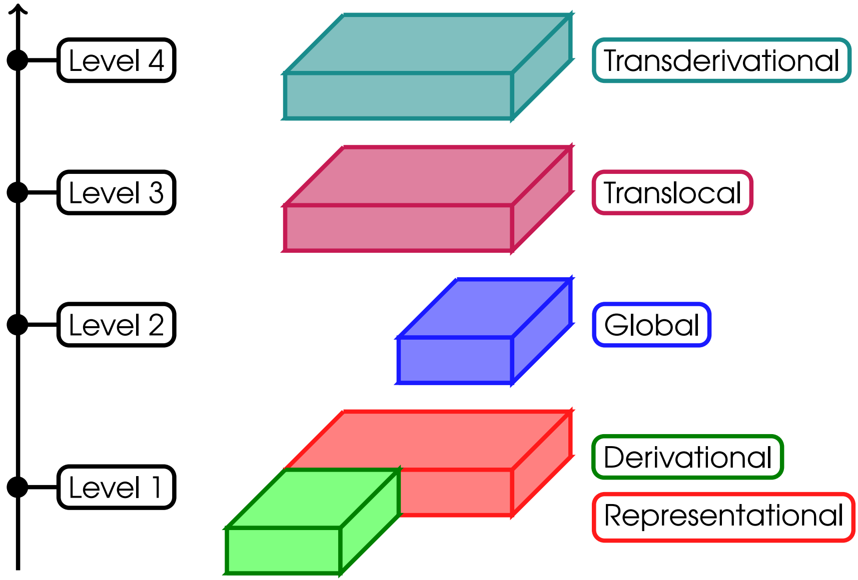
The Müller-Sternefeld-Hierarchy (Müller 2005)



The Müller-Sternefeld-Hierarchy (Müller 2005)



The Müller-Sternefeld-Hierarchy (Müller 2005)



Evaluation of the Hierarchy

Advantages

- Captures intuitions of syntacticians
- Relates constraints to locality

Shortcomings

- Framework specific
- Notion of complexity not well-defined
- No arguments for proposed order
- Hierarchy holds for classes or every single constraint?
- Counterexamples

But maybe the hierarchy isn't that far off the mark . . .

Evaluation of the Hierarchy

Advantages

- Captures intuitions of syntacticians
- Relates constraints to locality

Shortcomings

- Framework specific
- Notion of complexity not well-defined
- No arguments for proposed order
- Hierarchy holds for classes or every single constraint?
- Counterexamples

But maybe the hierarchy isn't that far off the mark . . .

Evaluation of the Hierarchy

Advantages

- Captures intuitions of syntacticians
- Relates constraints to locality

Shortcomings

- Framework specific
- Notion of complexity not well-defined
- No arguments for proposed order
- Hierarchy holds for classes or every single constraint?
- Counterexamples

But maybe the hierarchy isn't that far off the mark . . .

Outline

Why Multi-Dimensional Trees, and What are They?

Multi-Dimensional Trees — The Basic Idea (Rogers 2003)

- String = set ordered by precedence
- 2-tree = set ordered by precedence & dominance
- 3-tree = set ordered by precedence, dominance & 3-dominance
- ...

Advantages of Multi-Dimensional Tree Framework

- model-theoretic perspective natural choice for study of constraints
- allows encoding of popular frameworks (TAG, GPSG, GB, probably Minimalism and HPSG)

Why Multi-Dimensional Trees, and What are They?

Multi-Dimensional Trees — The Basic Idea (Rogers 2003)

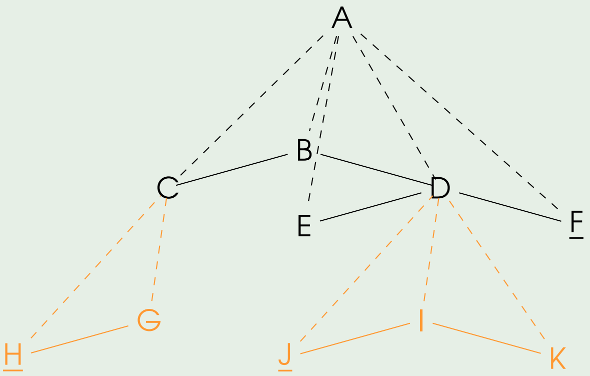
- String = set ordered by precedence
- 2-tree = set ordered by precedence & dominance
- 3-tree = set ordered by precedence, dominance & 3-dominance
- ...

Advantages of Multi-Dimensional Tree Framework

- model-theoretic perspective natural choice for study of constraints
- allows encoding of popular frameworks (TAG, GPSG, GB, probably Minimalism and HPSG)

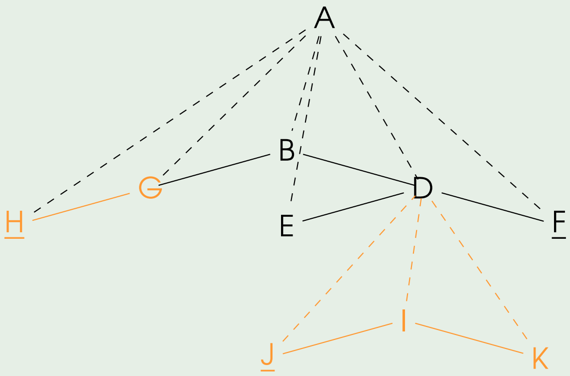
Examples

Example (3-Tree Representation of a TAG Derivation)



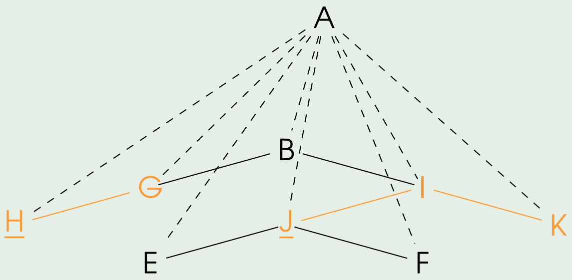
Examples

Example (3-Tree Representation of a TAG Derivation)



Examples

Example (3-Tree Representation of a TAG Derivation)



A Logic for Multi-Dimensional Trees

MSO^d — Monadic Second-Order Logic over d-Trees

MSO^d includes

- the boolean connectives,
- the usual grouping symbols,
- $\exists$ and $\forall$,
- a countably infinite set of variables over individuals,
- a countably infinite set of variables over finite subsets,
- a constant $\triangleleft_i$ for every reflexive i-dominance relation, $1 \leq i \leq d$.

Recognizable Sets

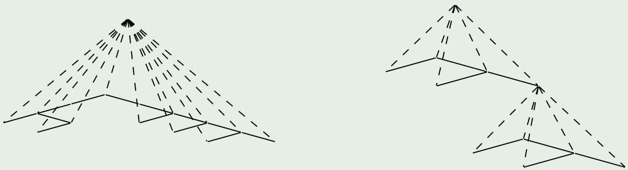
A set of d-trees is recognizable iff it is definable in MSO^d.

Local Sets

Local Trees

A *d*-tree is local iff it has depth 1 at dimension *d*.

Example (Local & Non-Local 3-Trees)



Local Sets

A recognizable set of *d*-trees is local iff it can be obtained from the composition of local trees iff it satisfies the subtree substitution property.

Local Sets

Local Trees

A d-tree is local iff it has depth 1 at dimension d.

Example (Local & Non-Local 3-Trees)

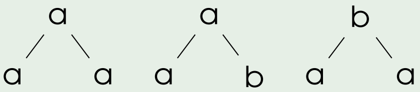
Local Sets

A recognizable set of d-trees is local iff it can be obtained from the composition of local trees iff it satisfies the subtree substitution property.

Example — Local and Recognizable Sets

Example (A Recognizable Set)

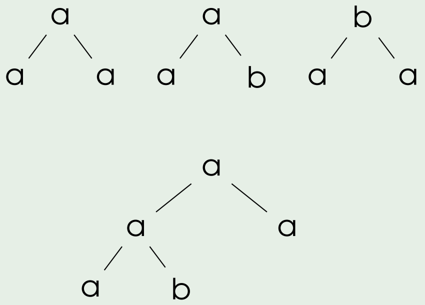
The set of $\{a, b\}$-labeled 2-trees containing exactly one node labeled b is not local, but recognizable.



Example — Local and Recognizable Sets

Example (A Recognizable Set)

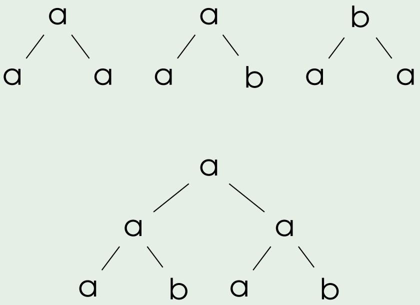
The set of $\{a, b\}$-labeled 2-trees containing exactly one node labeled b is not local, but recognizable.



Example — Local and Recognizable Sets

Example (A Recognizable Set)

The set of $\{a, b\}$-labeled 2-trees containing exactly one node labeled b is not local, but recognizable.



Outline

The Logic LOC^k for Local k-Dimensional Trees

LOC^k — A Logic for Local Sets of k-Trees

Turn MSO^k into a "modal" logic that cannot see beyond depth 1 in the main dimension by

- restricting quantification, and
- restricting quantifier scope at dimension k.

$RLOC^k$ — Relaxed LOC^k

The extension of LOC^k that can see beyond depth 1 in the main dimension.

Lemma ($RLOC^k \leq LOC^{k+1}$)

An MSO formula ϕ is an $RLOC^k$ formula iff it is a LOC^{k+1} formula containing no instances of $\triangleleft_{k+1}$.

The Logic LOC^k for Local k-Dimensional Trees

LOC^k — A Logic for Local Sets of k-Trees

Turn MSO^k into a “modal” logic that cannot see beyond depth 1 in the main dimension by

- restricting quantification, and
- restricting quantifier scope at dimension k.

$RLOC^k$ — Relaxed LOC^k

The extension of LOC^k that can see beyond depth 1 in the main dimension.

Lemma ($RLOC^k \leq LOC^{k+1}$)

An MSO formula ϕ is an $RLOC^k$ formula iff it is a LOC^{k+1} formula containing no instances of $\triangleleft_{k+1}$.

The Logic LOC^k for Local k-Dimensional Trees

LOC^k — A Logic for Local Sets of k-Trees

Turn MSO^k into a "modal" logic that cannot see beyond depth 1 in the main dimension by

- restricting quantification, and
- restricting quantifier scope at dimension k.

$RLOC^k$ — Relaxed LOC^k

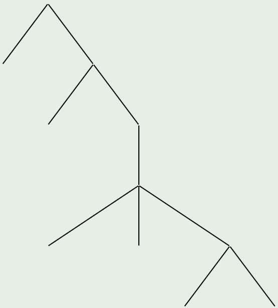
The extension of LOC^k that can see beyond depth 1 in the main dimension.

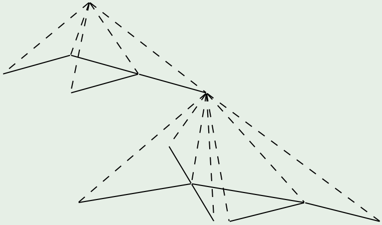
Lemma ($RLOC^k \leq LOC^{k+1}$)

An MSO formula ϕ is an $RLOC^k$ formula iff it is a LOC^{k+1} formula containing no instances of $\triangleleft_{k+1}$.

Examples of *LOC* and *RLOC* Constraints

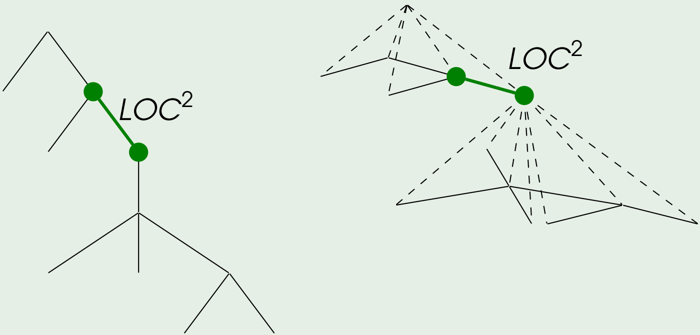
Example (*LOC* and *RLOC* Definable Constraints)





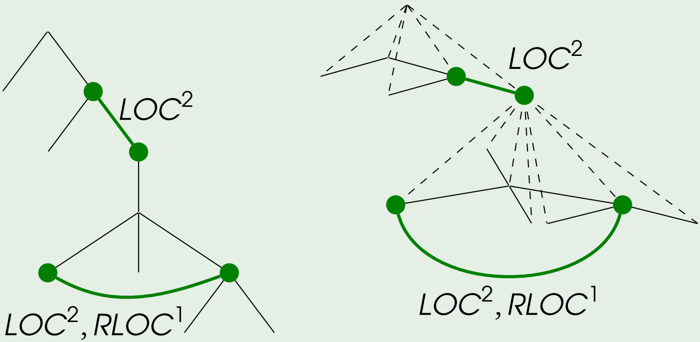
Examples of *LOC* and *RLOC* Constraints

Example (*LOC* and *RLOC* Definable Constraints)



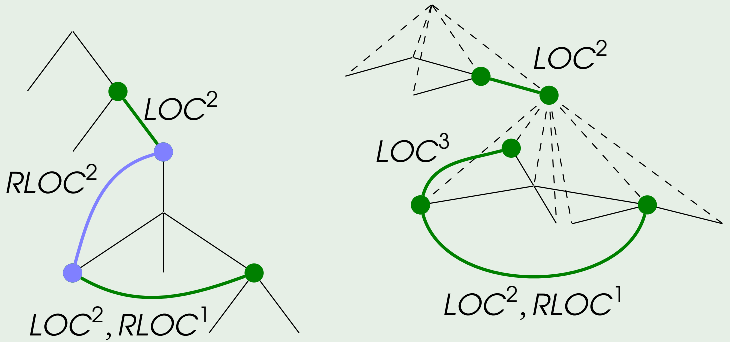
Examples of *LOC* and *RLOC* Constraints

Example (*LOC* and *RLOC* Definable Constraints)



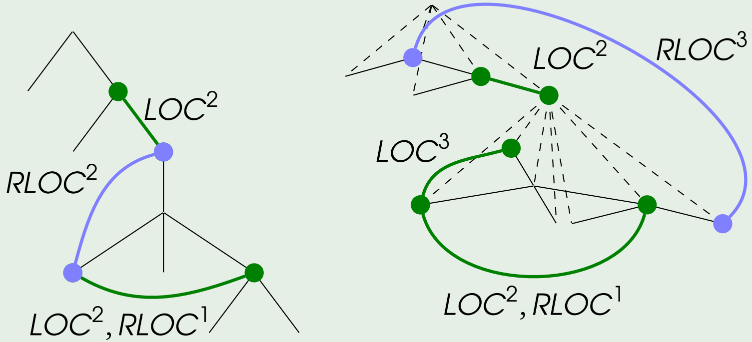
Examples of *LOC* and *RLOC* Constraints

Example (*LOC* and *RLOC* Definable Constraints)



Examples of *LOC* and *RLOC* Constraints

Example (*LOC* and *RLOC* Definable Constraints)

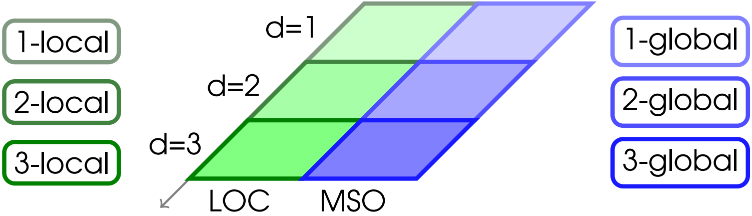


Definitions of Non-Comparative Constraints

Definition (Constraint Classes)

A constraint is

- *k*-global iff it is definable in MSO^k.
- *k*-local iff it is definable in LOC^k.
- fully *k*-local iff it is *k*-local and its respective LOC^k formula contains no MSO^i subformula ϕ, $0 < i \leq k$, such that $\phi \notin LOC^i$.



Reducibility Given a Variable Set of Features

Theorem (Global to Fully *k*-Local)

Let Φ be a set of MSO^d formulas and c_g a k-global constraint, $k \leq d$, such that $\Phi \cup \{c_g\}$ defines a recognizable set R. Then there is a fully k-local constraint c_l such that R is a projection of the set defined by $\Phi \cup \{c_l\}$.

Proof.

Follows from Thatcher's theorem that every recognizable set is a projection of a local set. That is, every recognizable set can be turned into a local one if we increase the number of features. □

Reducibility Given a Variable Set of Features

Theorem (Global to Fully k-Local)

Let Φ be a set of MSO^d formulas and c_g a k-global constraint, $k \leq d$, such that $\Phi \cup \{c_g\}$ defines a recognizable set R. Then there is a fully k-local constraint c_l such that R is a projection of the set defined by $\Phi \cup \{c_l\}$.

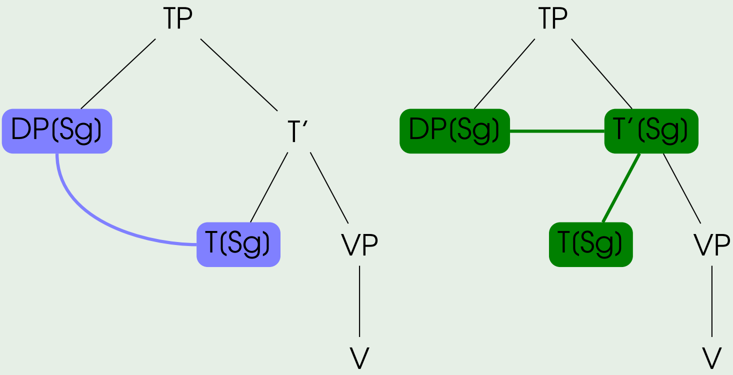
Proof.

Follows from Thatcher's theorem that every recognizable set is a projection of a local set. That is, every recognizable set can be turned into a local one if we increase the number of features. □

An Example of Feature Coding

Example (Feature Coding)

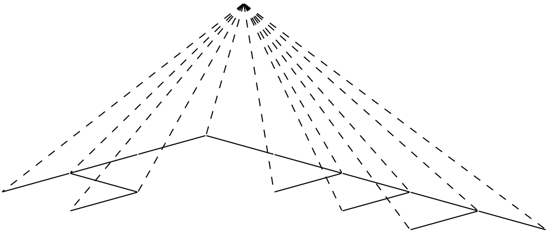
Replacing a 2-global agree constraint (left) by two 2-local ones (right) by adding new features:



Reducibility Given a Fixed Set of Features

Reducing Constraints Without Feature Coding

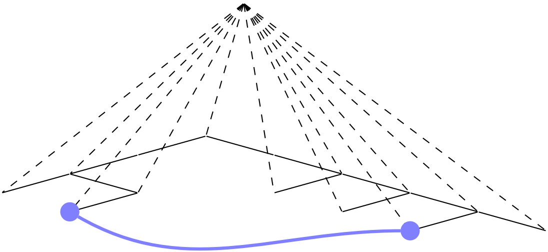
Weaker constraints at higher dimensions can replace stronger constraints at lower dimensions.
To make this precise, we study the expressivity of LOC^{k+1} with respect to $RLOC^{k}$ and MSO^{k}.



Reducibility Given a Fixed Set of Features

Reducing Constraints Without Feature Coding

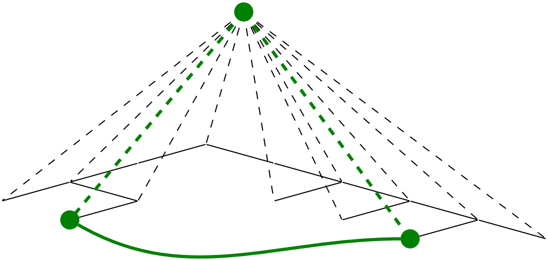
Weaker constraints at higher dimensions can replace stronger constraints at lower dimensions.
To make this precise, we study the expressivity of LOC^{k+1} with respect to $RLOC^k$ and MSO^k.



Reducibility Given a Fixed Set of Features

Reducing Constraints Without Feature Coding

Weaker constraints at higher dimensions can replace stronger constraints at lower dimensions.
To make this precise, we study the expressivity of LOC^{k+1} with respect to $RLOC^{k}$ and MSO^{k}.



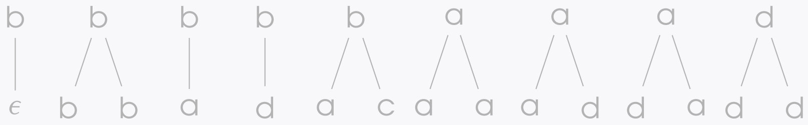
$RLOC^k < LOC^{k+1}$

Lemma ($RLOC^k < LOC^{k+1}$)

There is a set Φ of LOC^k formulas over labeled k-dimensional trees, $k > 1$, such that the $k-1$-dimensional yield of the tree language defined by Φ cannot be defined in $RLOC^{k-1}$.

Proof.

Consider $L := (\{a,b,d\}^* (a\{a,d\}^* c)^* \{a,b,d\}^*)^*$. This language cannot be defined in FO_2 and hence not in $RLOC^1$ either. The LOC^2 grammar below defines L:



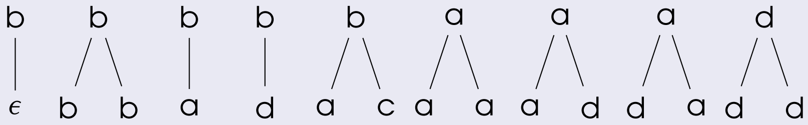
$RLOC^k < LOC^{k+1}$

Lemma ($RLOC^k < LOC^{k+1}$)

There is a set Φ of LOC^k formulas over labeled k-dimensional trees, $k > 1$, such that the $k-1$-dimensional yield of the tree language defined by Φ cannot be defined in $RLOC^{k-1}$.

Proof.

Consider $L := (\{a,b,d\}^* (a\{a,d\}^* c)^* \{a,b,d\}^*)^*$. This language cannot be defined in FO_2 and hence not in $RLOC^1$ either. The LOC^2 grammar below defines L:



$MSO^k \nleq LOC^{k+1}$

Lemma ($MSO^k \nleq LOC^{k+1}$)

There is a set Φ of MSO^k formulas over labeled k-dimensional trees, $k \geq 1$, such that there is no LOC^{k+1} definable tree language whose k-dimensional yield is identical to the tree language defined by Φ.

Proof.

Consider $L := (aa)^*$, which is definable in MSO^1 but not in FO and thus not in LOC^1 either. It is easy to show that it cannot be defined in LOC^2 by invoking the subtree substitution closure property of local sets. □

$MSO^k \nleq LOC^{k+1}$

Lemma ($MSO^k \nleq LOC^{k+1}$)

There is a set Φ of MSO^k formulas over labeled k-dimensional trees, $k \geq 1$, such that there is no LOC^{k+1} definable tree language whose k-dimensional yield is identical to the tree language defined by Φ.

Proof.

Consider $L := (aa)^*$, which is definable in MSO^1 but not in FO and thus not in LOC^1 either. It is easy to show that it cannot be defined in LOC^2 by invoking the subtree substitution closure property of local sets. □

Reducibility for $k < d$, Feature Set Fixed

Theorem (k-Global to $k+1$-Local)

Let Φ be a set of MSO^d formulas and $\mathbb{C}$ the set of all k-global but not k-local constraints c_g, $k < d$, such that $\Phi \cup \{c_g\}$ defines a recognizable set. Then some but not all $c_g \in \mathbb{C}$ can be replaced by a $k+1$-local constraint.

Proof.

- $LOC^k < RLOC^k < LOC^{k+1}$ and $RLOC^k < MSO^k$ entail existence of k-global but not k-local constraints definable in LOC^{k+1}
- $MSO^k \not\leq LOC^{k+1}$ shows that this does not hold for all k-global constraints □

Reducibility for $k < d$, Feature Set Fixed

Theorem (k-Global to $k+1$-Local)

Let Φ be a set of MSO^d formulas and $\mathbb{C}$ the set of all k-global but not k-local constraints c_g, $k < d$, such that $\Phi \cup \{c_g\}$ defines a recognizable set. Then some but not all $c_g \in \mathbb{C}$ can be replaced by a $k+1$-local constraint.

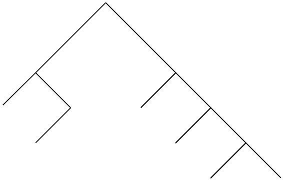
Proof.

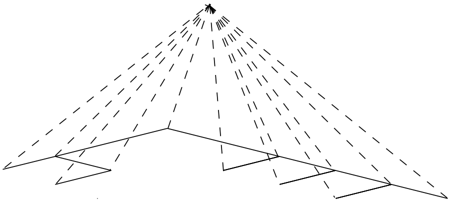
- $LOC^k < RLOC^k < LOC^{k+1}$ and $RLOC^k < MSO^k$ entail existence of k-global but not k-local constraints definable in LOC^{k+1}
- $MSO^k \nleq LOC^{k+1}$ shows that this does not hold for all k-global constraints □

Extension to $k = d$

Extending the Results to $k = d$

“Create more space”: Given a recognizable set R, add a $d + 1$-dimensional root to each of its members and limit the depth of $d + 1$-trees to 1.





Relevance to Linguistics

Relevance of Reducibility With Variable Feature Set

- Argues against use of feature coding mechanisms (Slash-feature, pied-piping, . . .)
- Recent attempts to reduce size of locality domain by use of diacritic features may fail to produce new insights

Relevance of Reducibility With Fixed Feature Set

- Different theories may use different notions of locality $\rightarrow$ be careful with comparisons!
- Recent attempts to reduce size of locality domain by use of derivational constraints can be reinterpreted as study of the subclass of 2-global constraints active in natural language.

Relevance to Linguistics

Relevance of Reducibility With Variable Feature Set

- Argues against use of feature coding mechanisms (Slash-feature, pied-piping, ...)
- Recent attempts to reduce size of locality domain by use of diacritic features may fail to produce new insights

Relevance of Reducibility With Fixed Feature Set

- Different theories may use different notions of locality → be careful with comparisons!
- Recent attempts to reduce size of locality domain by use of derivational constraints can be reinterpreted as study of the subclass of 2-global constraints active in natural language.

Outline

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1				
Output_2				
Output_3				
Output_4				
Output_5				
Output_6				

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1				
Output_2				
Output_3				
Output_4				
Output_5				
Output_6				

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1				
Output_2				
Output_3				
Output_4				
Output_5				
Output_6	*			

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1				
Output_2				
Output_3				
Output_4				
Output_5				

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1				
Output_2				
Output_3				
Output_4		*		
Output_5		*		

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output$_1$				
Output$_2$				
Output$_3$				

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1			*	
Output_2			*	
Output_3			**	

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output_1			*	
Output_2			*	

The Problem With Comparative Constraints

The Paradox of Comparative Constraints

For a comparative constraint to compute the best tree in some set of trees, all competing trees have to be members of this set. But this is ruled out by the constraint itself.

⇒ Optimality Systems (Frank and Satta 1998)

Example (Optimality System)

Input	C_1	C_2	C_3	C_4
Output$_1$			*	*
Output$_2$			*	*

Regularity of Optimality Systems (Jäger 2002)

Theorem (Optimality Systems as Regular Relations)

Let O be an optimality system such that

- the mapping R from inputs to outputs is a regular relation, and
- given input i, every constraint defines a regular relation S on the set of output candidates for i, and
- optimality is global with respect to R and S: if output o is optimal for input i, then o is also optimal with respect to the set of all possible outputs, regardless of the input.

Then O defines a regular relation.

Application to Constraints

Corollary (Comparative to Global)

A proper subclass of the class of comparative constraints can be reduced to global constraints.

Relevance to Syntax

For all syntactic comparative constraints, the inverse of the input-output mapping is a function, whence optimality is global. Therefore, the reducability of these constraints depends solely on their definition of reference set.

Approaches for Semantics & Pragmatics

- Bidirectional optimality systems
- Bimorphisms
- Category theory (higher-order optimality systems)

Application to Constraints

Corollary (Comparative to Global)

A proper subclass of the class of comparative constraints can be reduced to global constraints.

Relevance to Syntax

For all syntactic comparative constraints, the inverse of the input-output mapping is a function, whence optimality is global. Therefore, the reducability of these constraints depends solely on their definition of reference set.

Approaches for Semantics & Pragmatics

- Bidirectional optimality systems
- Bimorphisms
- Category theory (higher-order optimality systems)

Application to Constraints

Corollary (Comparative to Global)

A proper subclass of the class of comparative constraints can be reduced to global constraints.

Relevance to Syntax

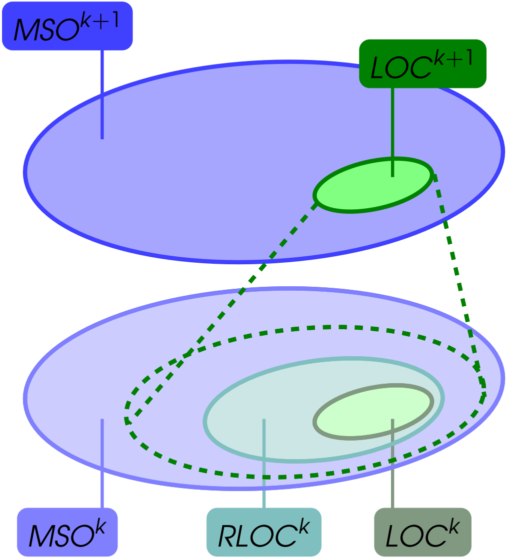
For all syntactic comparative constraints, the inverse of the input-output mapping is a function, whence optimality is global. Therefore, the reducability of these constraints depends solely on their definition of reference set.

Approaches for Semantics & Pragmatics

- Bidirectional optimality systems
- Bimorphisms
- Category theory (higher-order optimality systems)

Conclusion

- Multi-dimensional trees as grammar neutral framework for study of constraints
- Constraints reducible under specific conditions
- Big picture of Müller-Sternefeld-hierarchy confirmed



Frank, Robert, and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Jäger, Gerhard. 2002. Gradient constraint in finite state OT: The unidirectional and the bidirectional case. In *More than words. a festschrift for dieter wunderlich*, ed. I. Kaufmann and B. Stiebels, 299–325. Berlin: Akademie Verlag.

Müller, Gereon. 2005. Constraints in syntax. Lecture Notes, Universität Leipzig.

Müller, Gereon, and Wolfgang Sternefeld. 2000. The rise of competition in syntax: A synopsis. In *Competition in syntax*, ed. Wolfgang Sternefeld and Gereon Müller, 1–68. Berlin: Mouton de Gruyter.

Rogers, James. 2003. Syntactic structures as multi-dimensional trees. *Research on Language and Computation* 1:265–305.

Thatcher, J. W. 1967. Characterizing derivation trees for context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences* 1:317–322.