Movement-Generalized Minimalist Grammars

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

University of California, Los Angeles

LACL 2012
July 2, 2012

Outline

MSO as a Graph Description Language

- MSO $=$ extension of FO with quantification over sets
- A set L of graphs is MSO-definable iff there is an MSO-formula ϕ such that L is the set of models of ϕ.
- A tree language is regular iff it is MSO-definable.

Common Predicates

Symbol	**Relation**
$\triangleleft$	immediate dominance
$\triangleleft^+$	proper dominance
$\approx$	equivalence
σ	label σ

MSO Graph Transductions

Basic Idea

Interpret output graph as (sub)structure of the input graph

Definition (MSO graph transduction)

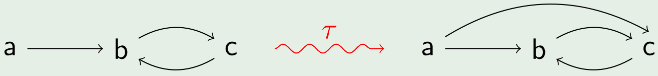
A (non-copying) **MSO graph transduction** τ consists of

- an MSO formula ϕ defining the domain of the input graph,
- an MSO formula ψ defining the domain of the output graph,
- a family of MSO formulas defining the relations in the output graph in terms of the relations in the input graph.

A graph transduction from trees to trees is called an MSO tree transduction (MSO-TT).

Examples

Two Easy Transductions



Edge relation: $\triangleleft$ Edge relation: $\blacktriangleleft$

$$\tau : x \blacktriangleleft y \leftrightarrow x \triangleleft^{+} y \wedge x \not\approx y$$

Edge relation: $\triangleleft$ Edge relation: $\blacktriangleleft$

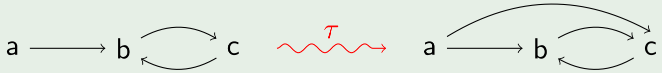
$$\tau : x \blacktriangleleft y \leftrightarrow \exists z[x \triangleleft z \wedge y \triangleleft z]$$

$$d(x) \leftrightarrow a(x) \vee c(x)$$

$$b(x) \leftrightarrow b(x)$$

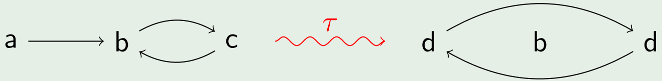
Examples

Two Easy Transductions



Edge relation: $\lhd$ Edge relation: $\blacktriangleleft$

$$\tau : x \blacktriangleleft y \leftrightarrow x \lhd^{+} y \wedge x \not\approx y$$



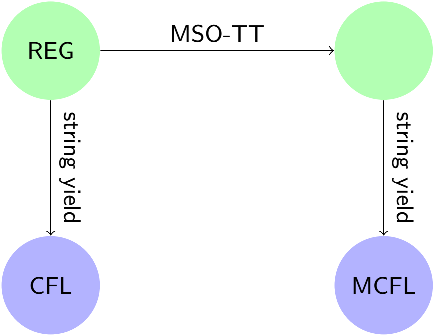
Edge relation: $\lhd$ Edge relation: $\blacktriangleleft$

$$\tau : x \blacktriangleleft y \leftrightarrow \exists z[x \lhd z \wedge y \lhd z]$$
$$d(x) \leftrightarrow a(x) \vee c(x)$$
$$b(x) \leftrightarrow b(x)$$

MSO and MCFLs

- Regular tree languages yield CFLs.
- However, the class of MCFLs is identical to the string yield of the image of the regular tree languages under MSO tree transductions (cf. Mönnich 2006).

MCFL ≡ str(MSO-TT(REG))



Minimalist Grammars

- Originally formulated in Stabler (1997)
- Highly lexicalized formalism inspired by Minimalist syntax (Chomsky 1995).
- Weakly equivalent to MCFGs (Harkema 2001; Michaelis 1998, 2001)
- More succinct than MCFGs (Stabler 2012)
- Derivation trees provide compact finite-state representation (Michaelis 2001; Kobele et al. 2007)
- Attractive closure properties (Graf 2011; Kobele 2011)
- Very extensible while preserving weak generative capacity

The Atoms of a Minimalist Grammar

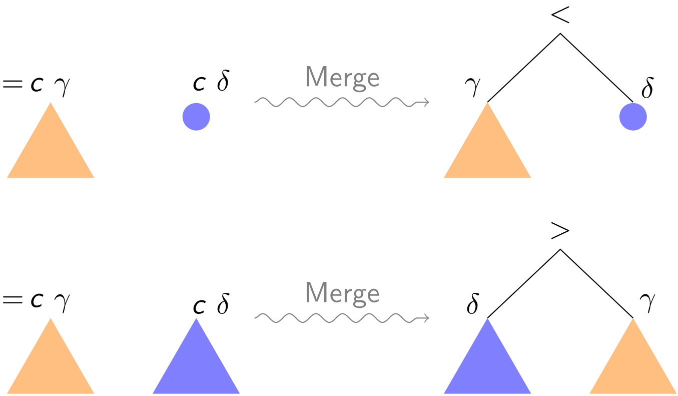
Minimalist Grammars (Stabler 1997)

An MG is a 5-tuple $G := \langle \Sigma, Feat, F, Lex, Op \rangle$, where

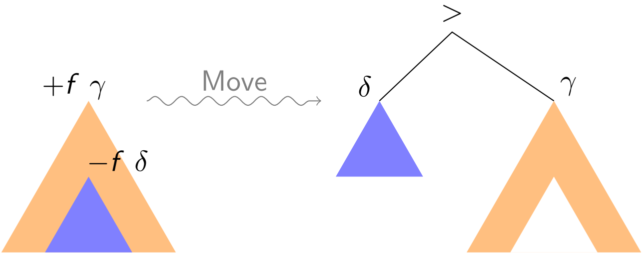
- Σ is an alphabet,
- *Feat* is a non-empty finite set of
  - category features f,
  - selector features $= f$,
  - movement licensee features $-f$,
  - movement licensor features $+f$,
- $F \subseteq Feat$ is a set of final category features,
- the lexicon *Lex* is a finite subset of $\Sigma^* \times Feat^+$,
- $Op := \{merge, move\}$ is the set of structure-building operations.

For every MGs it suffices to specify *Lex* and F.

Operation 1: Merge

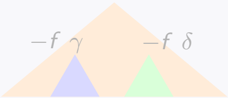


Operation 2: Move

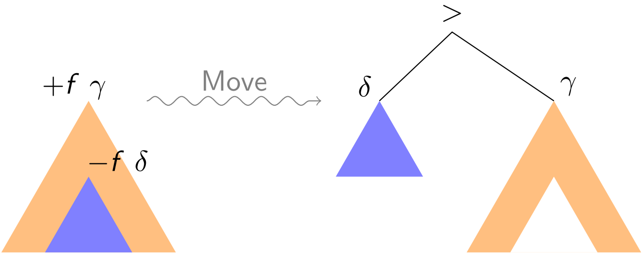


Shortest Move Constraint (SMC)

No two lexical items may have the same licensee feature as their first unchecked feature.

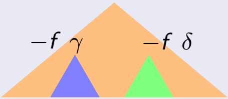


Operation 2: Move



Shortest Move Constraint (SMC)

No two lexical items may have the same licensee feature as their first unchecked feature.



A Toy Example (Without Recursion)

MG with $F = \{C\}$

men :: N	like :: $= \mathrm{D} \ = \mathrm{D} \ \mathrm{V}$
the :: $= \mathrm{N} \ \mathrm{D}$	ε :: $= \mathrm{V} \ \mathrm{C}$
which :: $= \mathrm{N} \ \mathrm{D} \ - \mathrm{wh}$	do :: $= \mathrm{V} \ + \mathrm{wh} \ \mathrm{C}$

A Toy Example (Without Recursion)

MG with $F = \{C\}$

men :: N	like :: $=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}$
the :: $=\mathrm{N}\ \mathrm{D}$	ε :: $=\mathrm{V}\ \mathrm{C}$
which :: $=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}$	do :: $=\mathrm{V}\ +\mathrm{wh}\ \mathrm{C}$

$$\frac{\text{the}}{=\mathrm{N}\ \mathrm{D}} \quad \frac{\text{men}}{\mathrm{N}} \quad \frac{\text{like}}{=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}} \quad \frac{\text{which}}{=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}} \quad \frac{\text{men}}{\mathrm{N}}$$

A Toy Example (Without Recursion)

MG with $F = \{C\}$

men :: N	like :: $=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}$
the :: $=\mathrm{N}\ \mathrm{D}$	ε :: $=\mathrm{V}\ \mathrm{C}$
which :: $=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}$	do :: $=\mathrm{V}\ +\mathrm{wh}\ \mathrm{C}$

A Toy Example (Without Recursion)

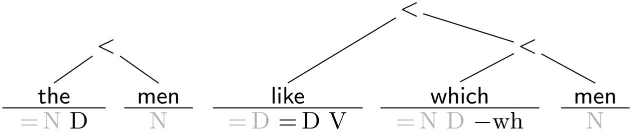
MG with $F = \{C\}$

men :: N	like :: $= \mathrm{D} \ = \mathrm{D} \ \mathrm{V}$
the :: $= \mathrm{N} \ \mathrm{D}$	ε :: $= \mathrm{V} \ \mathrm{C}$
which :: $= \mathrm{N} \ \mathrm{D} \ - \mathrm{wh}$	do :: $= \mathrm{V} \ + \mathrm{wh} \ \mathrm{C}$

A Toy Example (Without Recursion)

MG with $F = \{C\}$

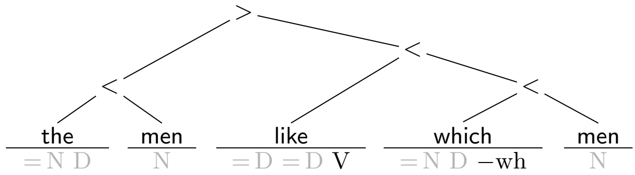
men :: N	like :: $=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}$
the :: $=\mathrm{N}\ \mathrm{D}$	ε :: $=\mathrm{V}\ \mathrm{C}$
which :: $=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}$	do :: $=\mathrm{V}\ +\mathrm{wh}\ \mathrm{C}$



A Toy Example (Without Recursion)

MG with $F = \{C\}$

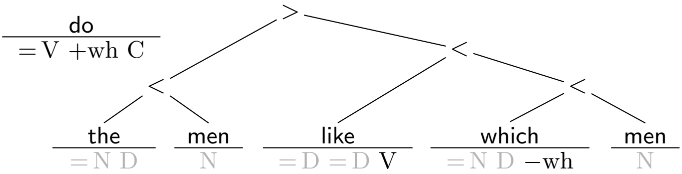
men :: N	like :: $= \mathrm{D} \ = \mathrm{D} \ \mathrm{V}$
the :: $= \mathrm{N} \ \mathrm{D}$	ε :: $= \mathrm{V} \ \mathrm{C}$
which :: $= \mathrm{N} \ \mathrm{D} \ - \mathrm{wh}$	do :: $= \mathrm{V} \ + \mathrm{wh} \ \mathrm{C}$



A Toy Example (Without Recursion)

MG with $F = \{C\}$

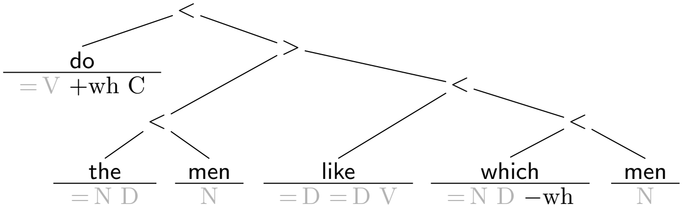
men :: N	like :: $=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}$
the :: $=\mathrm{N}\ \mathrm{D}$	ε :: $=\mathrm{V}\ \mathrm{C}$
which :: $=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}$	do :: $=\mathrm{V}\ +\mathrm{wh}\ \mathrm{C}$



A Toy Example (Without Recursion)

MG with $F = \{C\}$

men :: N	like :: $=\mathrm{D}\ =\mathrm{D}\ \mathrm{V}$
the :: $=\mathrm{N}\ \mathrm{D}$	ε :: $=\mathrm{V}\ \mathrm{C}$
which :: $=\mathrm{N}\ \mathrm{D}\ -\mathrm{wh}$	do :: $=\mathrm{V}\ +\mathrm{wh}\ \mathrm{C}$



A Toy Example (Without Recursion)

MG with $F = \{C\}$

men :: N

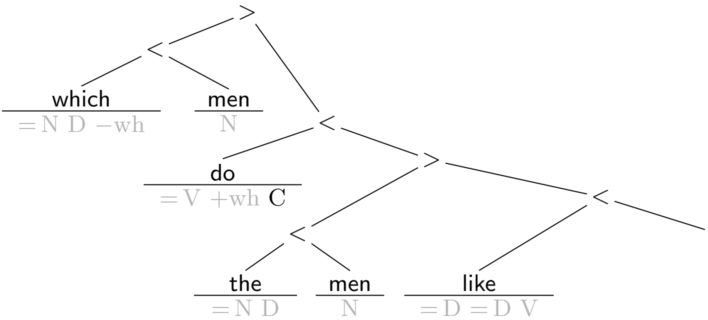
the :: $= \mathrm{N}\ \mathrm{D}$

which :: $= \mathrm{N}\ \mathrm{D}\ - \mathrm{wh}$

like :: $= \mathrm{D}\ = \mathrm{D}\ \mathrm{V}$

ε :: $= \mathrm{V}\ \mathrm{C}$

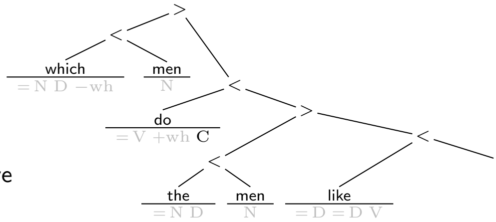
do :: $= \mathrm{V}\ + \mathrm{wh}\ \mathrm{C}$

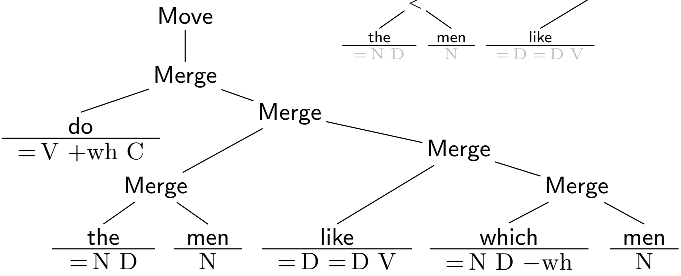


A Model-Theoretic View of MGs

- Every MG is uniquely specified by its set of derivations.
- Derivations are trees ⇒ can be defined model-theoretically
- Each node in a derivation "belongs" to some lexical item ⇒ lexicon is a finite set of treelets called **slices**
- A derivation tree is a combination of slices that respects the constraints of the feature calculus.
- A given MG's derivation tree language is the largest set of such well-formed slice combinations.
- Crucially, the constraints of the feature calculus can be **expressed in tree-geometric terms**.

Derivation Trees





Slices

Intuitively, slices are the **derivation tree equivalent of phrasal projection**: Each slice marks the subpart of the derivation that a lexical item has control over by virtue of its selector and licensor features.

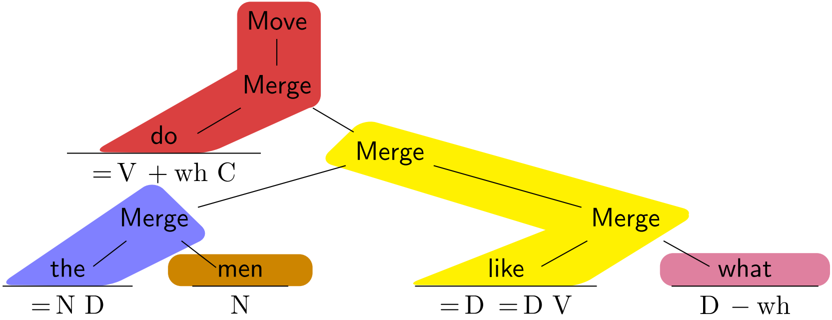
Slices

Given a derivation tree t and lexical item l occurring in t, $\mathrm{slice}(l)$ is defined as follows:

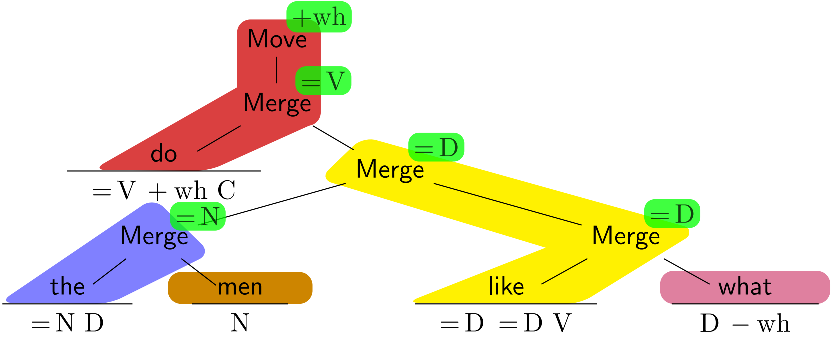
- $l \in \mathrm{slice}(l)$,
- if node n of t immediately dominates a node $s \in \mathrm{slice}(l)$, then $n \in \mathrm{slice}(l)$ iff the operation denoted by the label of n erased a selector or licensor feature of l.

The unique $n \in \mathrm{slice}(l)$ that isn't (properly) dominated by any $n' \in \mathrm{slice}(l)$ is called the *slice root* of l.

Example of Slices



Example of Slices



Regulating Merge

Merge

If n is a Merge node associated to selector feature $= f$, it immediately dominates the slice root of some lexical item with category feature f.

Regulating Move: Finding Occurrences

A Move node that checks a licensee feature of lexical item l is called an **occurrence** of l.

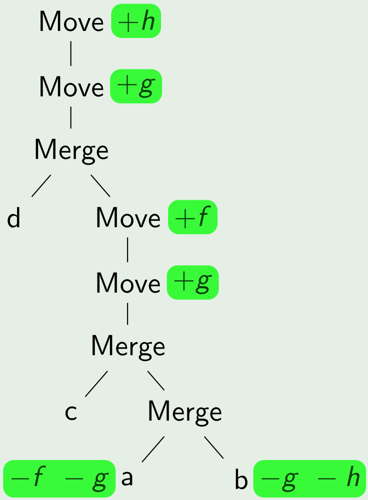
Definition (Occurrences)

For every combination t of slices and lexical item l in t with licensee features $-f_1 \cdots - f_n$, the **occurrences** of l in t are:

- $occ_0(l)$ is the mother of the slice root of l in t (if it exists).
- $occ_i(l)$ is the unique node m of t labeled Move such that
  - m is associated to $+f_i$, and
  - m **properly dominates** occ_{i-1}, and
  - there is no node n in t such that
    - n is associated to $+f_i$, and
    - n properly dominates occ_{i-1}, and
    - n is properly dominated by m.

Exercise: Find the Occurrences!

Example



Regulating Move: Resource Sensitivity & SMC

Move

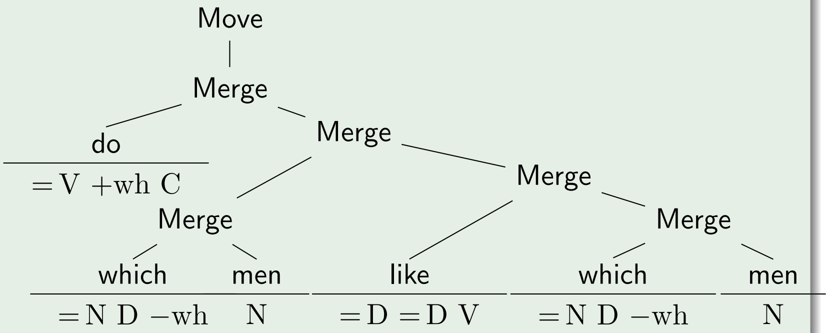
For every lexical item l with licensee features $-f_1 \cdots - f_n$, there exist nodes $m_1, \ldots, m_n$ such that m_i (and no other node) is the ith occurrence of l.

SMC

For every Move node m there is exactly one lexical item l such that m is an occurrence of l.

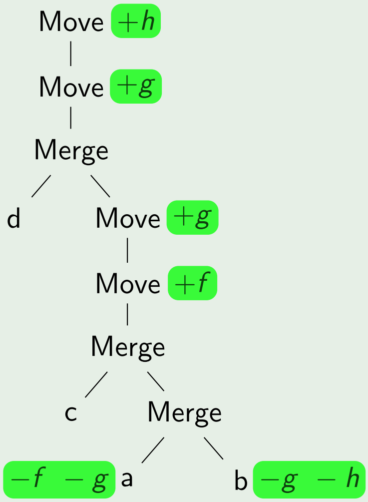
Why the SMC Works

Example 1



Why the SMC Works [cont.]

Example 2



Mapping to Derived Trees

- Derivation trees can be mapped to derived trees by a multi bottom-up transducer (Kobele et al. 2007)
- However, an MSO tree transduction is more intuitive.
- The transduction adds a dominance branch **from an LI's highest occurrence to the root of its slice**. The original dominance branch is removed.
- Linear order and interior node labels also need to be changed. See the paper for details.

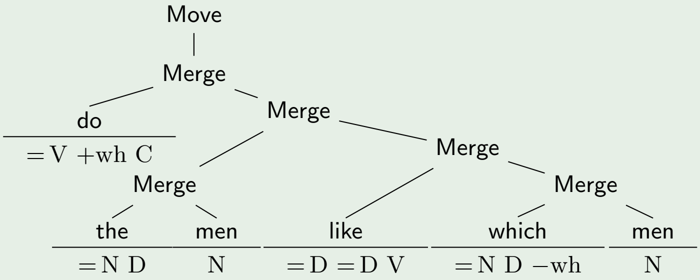
The Transduction Formula

$$x \blacktriangleleft y \leftrightarrow x \triangleleft y \vee \exists l[\textit{f-occ}(x, l) \wedge \textit{sliceroot}(y, l)]$$

$$\textit{f-occ}(x, l) \leftrightarrow \textit{occurrence}(x, l) \wedge \neg \exists z[z \triangleleft^{+} x \wedge \textit{occurrence}(z, l)]$$

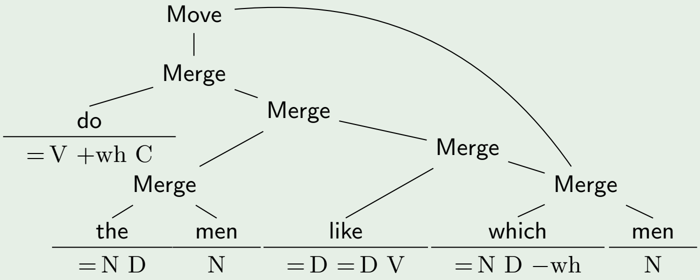
Example Transduction

Example



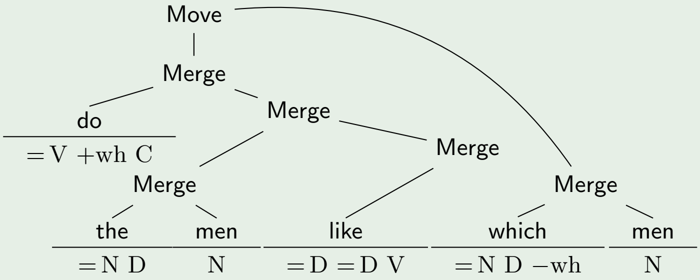
Example Transduction

Example



Example Transduction

Example



The Basic Idea

- We now have a FO-definable theory of Minimalist derivations and a simple MSO transduction to derived trees.
- But there is wiggle room: MGs generate MCFLs, and **MCFL = str(MSO-TT(REG))**.
- So we can change certain parameters in the definitions, as long as MSO-definability is preserved
  $\Rightarrow$ **parametric template to create new variants of Move**

Parameters of Move

The variants of Move discussed in the syntactic literature vary along several dimensions:

- Possible landing sites: raising, lowering, sidewards
- Size of moved constituent: phrase, head, pied-piped phrase
- Visibility: overt, covert
- Directionality: left, right

Loci of Parameters

Parameter	Locus
landing site	definition of occurrence
size	transduction
visibility	transduction
directionality	transduction

Replacing Proper Dominance

Reminder: Definition of Occurrence

$occ_i(l)$ is the unique node m of t labeled Move such that

- m is associated to $+f_i$, and
- m **properly dominates** occ_{i-1}, and
- there is no node n in t such that
  - n is associated to $+f_i$, and
  - n properly dominates occ_{i-1}, and
  - n is properly dominated by m.

Alternative Instantiations

Relation	**Movement Type**
inverse proper dominance	lowering
proper dominance across at most 1 slice	antilocal raising
slice containment	sidewards
slice containment restricted to wh-phrases	wh-clustering

Size of Moved Constituent

Reminder: The Transduction Formula

$$x \blacktriangleleft y \leftrightarrow x \triangleleft y \vee \exists l[\textit{f-occ}(x, l) \wedge \textbf{sliceroot}(y, l)]$$

Alternative Instantiations

Formula	**Movement Type**
$y \approx l$	head movement
$\exists l'[\textit{sliceroot}(y, l') \wedge \textit{selects}(l', l)]$	pied-piping

Further Observations

- With respect to directionality, all movement types can be **replaced by a combination of raising and lowering** (but new lexical items might be needed to furnish extra landing sites).
- In most cases, **strong generative capacity is increased** (e.g. lowering makes it possible to generate any given TAL with X′-like trees).
- Many MSO-definable relations do not yield useful movement types (e.g. all reflexive and all symmetric relations).

Conclusion

What was Accomplished?

- MSO definition of MGs as derivation tree languages with a mapping to derived trees.
- Definitions provide a parameterized template that can easily be tuned to create new movement types without increasing weak generative capacity.

Why Should You Care?

- Combined with the results of Graf (2011) and Kobele (2011), we now have a general system for adding constraints and movement types to MGs $\Rightarrow$ extremely flexible framework (just about anything from the literature can be incorporated)
- New movement types make it possible to emulate other formalisms $\Rightarrow$ resource sharing, new perspectives

References I

Chomsky, Noam. 1995. *The minimalist program*. Cambridge, Mass.: MIT Press.

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.

Harkema, Henk. 2001. A characterization of minimalist languages. In *Logical aspects of computational linguistics (lacl'01)*, ed. Philippe de Groote, Glyn Morrill, and Christian Retoré, volume 2099 of *Lecture Notes in Artificial Intelligence*, 193–211. Berlin: Springer.

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., Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10*, ed. James Rogers and Stephan Kepser, 71–80.

Michaelis, Jens. 1998. Derivational minimalism is mildly context-sensitive. *Lecture Notes in Artificial Intelligence* 2014:179–198.

Michaelis, Jens. 2001. Transforming linear context-free rewriting systems into minimalist grammars. *Lecture Notes in Artificial Intelligence* 2099:228–244.

Mönnich, Uwe. 2006. Grammar morphisms. Ms. University of Tübingen.

References II

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. 2012. Bayesian, minimalist, incremental syntactic analysis. Ms., UCLA.