

Eine Bemerkung zur Komplexität der Tilgungstheorie elliptischer Sätzen

Gregory M. Kobele

kobele@rz.hu-berlin.de

Institut für deutsche Sprache und Linguistik
Humboldt Universität zu Berlin

Workshop Theoretischer Computerlinguistik
Tübingen, 28.November 2008

The Big Picture

The question I am asking is:

What is the complexity of the form–meaning relation permitted by minimalist grammars extended so as to allow for deletion under identity?

My answer will be:

Every form–meaning relation definable by such MGs is representable as a triple $\langle f, T, m \rangle$, where f, m are homomorphisms from T to sounds and meanings respectively, and T is a set of trees definable in first order logic with a binary deterministic transitive closure operator ($FO(DTC^2)$).

The Big Picture

The question I am asking is:

What is the complexity of the form–meaning relation permitted by minimalist grammars extended so as to allow for deletion under identity?

My answer will be:

Every form–meaning relation definable by such MGs is representable as a triple $\langle f, T, m \rangle$, where f, m are homomorphisms from T to sounds and meanings respectively, and T is a set of trees definable in first order logic with a binary deterministic transitive closure operator ($FO(DTC^2)$).

- 1 The Basic Problem
- 2 Minimalist Grammars
- 3 Ellipsis
- 4 Derivations with Delete

- At one level of description, a language can be viewed as a set of pairs

$$\{\langle s, m \rangle : s \in \text{Sound} \ \& \ m \in \text{Meaning}\}$$

- **The** question: what properties does the set \mathcal{L} of all possible human languages have?
- A(n immediate) problem: What is in the set Meaning?
 - Truth/Verification conditions
 - Descriptions of use
 - Algorithms/Logic programs
 - Structured objects suitable for computing entailments

Representing Meanings

- I will take as a legitimate 'meaning representation' anything which can be compositionally interpreted as a model-theoretic object.
- an expression's meaning will be represented as its *derivation tree*.
 - In 'standard' GB/Minimalism, this is not thought appropriate: compositional semantic interpretation is traditionally defined over LF, not the derivation tree (LF is at best a homomorphic image of a tree of which the derivation tree is a projection)
 - However, I have shown that canonical examples of phenomena which have been used to support the LF-Interpretation Hypothesis (scope inversion and inverse linking) can be accounted for directly in terms of the derivation tree (or a tree which projects the derivation tree)
- Thus I am reframing the question of the complexity of the allowable form-meaning relations to the complexity of the allowable form-*derivation tree* relations:

$$\{\langle s, m \rangle : s \in \text{Sound} \ \& \ m \in \text{DerivationTree}\}$$

Representing Meanings

- I will take as a legitimate 'meaning representation' anything which can be compositionally interpreted as a model-theoretic object.
- an expression's meaning will be represented as its *derivation tree*.
 - In 'standard' GB/Minimalism, this is not thought appropriate: compositional semantic interpretation is traditionally defined over LF, not the derivation tree (LF is at best a homomorphic image of a tree of which the derivation tree is a projection)
 - However, I have shown that canonical examples of phenomena which have been used to support the LF-Interpretation Hypothesis (scope inversion and inverse linking) can be accounted for directly in terms of the derivation tree (or a tree which projects the derivation tree)
- Thus I am reframing the question of the complexity of the allowable form-meaning relations to the complexity of the allowable form-*derivation tree* relations:

$$\{\langle s, m \rangle : s \in \text{Sound} \ \& \ m \in \text{DerivationTree}\}$$

Representing Meanings

- I will take as a legitimate 'meaning representation' anything which can be compositionally interpreted as a model-theoretic object.
- an expression's meaning will be represented as its *derivation tree*.
 - In 'standard' GB/Minimalism, this is not thought appropriate: compositional semantic interpretation is traditionally defined over LF, not the derivation tree (LF is at best a homomorphic image of a tree of which the derivation tree is a projection)
 - However, I have shown that canonical examples of phenomena which have been used to support the LF-Interpretation Hypothesis (scope inversion and inverse linking) can be accounted for directly in terms of the derivation tree (or a tree which projects the derivation tree)
- Thus I am reframing the question of the complexity of the allowable form-meaning relations to the complexity of the allowable form-*derivation tree* relations:

$$\{\langle s, m \rangle : s \in \text{Sound} \ \& \ m \in \text{DerivationTree}\}$$

- 1 The Basic Problem
- 2 Minimalist Grammars**
- 3 Ellipsis
- 4 Derivations with Delete

Minimalist Grammars

Minimalist grammars [Stabler, 1997] provide a formal perspective on some fundamental aspects of minimalism. A minimalist grammar is given by a four-tuple $\langle V, Cat, Lex, \mathcal{F} \rangle$, where

- V , the alphabet, is a finite non-empty set
- Cat , a finite set of features of the form

$$\begin{pmatrix} +f & =f \\ -f & f \end{pmatrix}$$

- Lex , the lexicon, is a finite set of pairs $\langle v, \delta \rangle$, for $v \in V \cup \{\epsilon\}$, and $\delta \in Cat^*$
- $\mathcal{F} = \{\text{MERGE}, \text{MOVE}\}$ is the set of structure building operations

| | | | |
|---|---------------------------|---|--------------------------|
| | John kisses Mary | * | John gives Mary |
| * | John kisses Mary the book | | John gives Mary the book |

- Words have requirements that must be satisfied

- *gives* selects three noun phrases

$\langle \text{gives}, =d =d =d s \rangle$

- *Bill* needs to be selected as a noun phrase

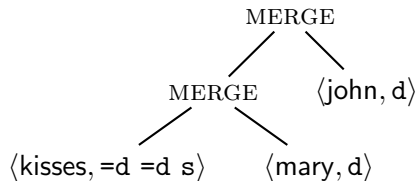
$\langle \text{Bill}, d \rangle$

- MERGE is a binary operation which enables such requirements to be met.

$\text{MERGE}(\langle \text{kisses}, =d =d s \rangle, \langle \text{mary}, d \rangle) = \langle \text{kisses mary}, =d s \rangle$

$\text{MERGE}(\langle \text{kisses mary}, =d s \rangle, \langle \text{john}, d \rangle) = \langle \text{john kisses mary}, s \rangle$

* $\text{MERGE}(\langle \text{john kisses mary}, s \rangle, \langle \text{bill}, d \rangle)$



| |
|----------------------------|
| John kissed Mary |
| Who did John kiss |
| *Who did John kiss Mary |
| *Mary did John kiss |

- Words can satisfy requirements of multiple others.
 - *who* is selected by *kiss*
 - and is later re-used to satisfy *did*

$\langle \text{who}, d -w \rangle$

$\langle \text{did}, =s +w q \rangle$

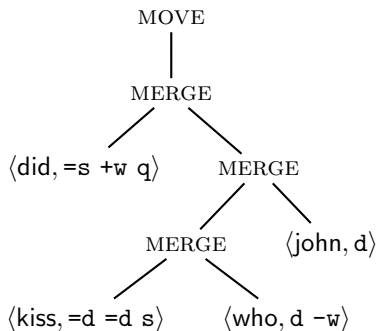
Minimalist Grammars

- MOVE is a unary operation which enables such requirements to be met.

$\text{MERGE}(\langle \text{kiss}, =d =d s \rangle; \langle \text{who}, d -w \rangle) = \langle \text{kiss}, =d s \rangle, \langle \text{who}, -w \rangle$

$\text{MERGE}(\langle \text{did}, =s +w q \rangle; \langle \text{john kiss}, s \rangle, \langle \text{who}, w \rangle) = \langle \text{did john kiss}, +w q \rangle, \langle \text{who}, -w \rangle$

$\text{MOVE}(\langle \text{did john kiss}, +w q \rangle, \langle \text{who}, -w \rangle) = \langle \text{who did john kiss}, q \rangle$



- It has long been observed that movement cannot relate arbitrary elements, but rather that there are constraints on which positions a moved item can be construed as originating from.
- The canonical constraint on movement in minimalist grammars is the SMC [Stabler, 1997].

The SMC

If an expression can move, it must move

- The SMC is implemented as a restriction on the domain of MOVE:
 $\text{MOVE}(\langle \alpha, +\mathbf{x}\gamma \rangle, \phi_1, \dots, \langle \beta, -\mathbf{x}\delta \rangle, \dots, \phi_n)$ is defined iff no other ϕ_i is of the form $\langle \eta, -\mathbf{x}\zeta \rangle$

A simple example

- $V = \{\text{seem, laugh, Mary, to, will}\}$

- $Cat = \{=i, i, =v, v, =d, d, s, +k, -k\}$

- $Lex = \left\{ \begin{array}{ll} \langle \text{seem, =i v} \rangle & \langle \text{laugh, =d v} \rangle \\ & \langle \text{Mary, d -k} \rangle \\ \langle \text{to, =v i} \rangle & \langle \text{will, =v +k s} \rangle \end{array} \right\}$

A simple grammar

| | |
|---------------------------------------|--|
| $\langle \text{seem}, =i \ v \rangle$ | $\langle \text{laugh}, =d \ v \rangle$ |
| $\langle \text{to}, =v \ i \rangle$ | $\langle \text{Mary}, d \ -k \rangle$ |
| | $\langle \text{will}, =v \ +k \ s \rangle$ |

- With the lexicon above, we can derive the language:

Mary will laugh
Mary will seem to laugh
⋮

An example derivation

- 1 MERGE($\langle \text{laugh}, =d \ v \rangle$, $\langle \text{Mary}, d \ -k \rangle$)
- 2 MERGE($\langle \text{will}, =v \ +k \ s \rangle$, 1)
- 3 MOVE(2)

$\langle \text{laugh}, v \rangle$, $\langle \text{Mary}, -k \rangle$ $\langle \text{will laugh}, +k \ s \rangle$, $\langle \text{Mary}, -k \rangle$
 $\langle \text{Mary will laugh}, s \rangle$

An example derivation

- 1 MERGE($\langle \text{laugh}, =d \ v \rangle$, $\langle \text{Mary}, d \ -k \rangle$)
- 2 MERGE($\langle \text{will}, =v \ +k \ s \rangle$, 1)
- 3 MOVE(2)

$\langle \text{laugh}, v \rangle$, $\langle \text{Mary}, -k \rangle$ $\langle \text{will laugh}, +k \ s \rangle$, $\langle \text{Mary}, -k \rangle$
 $\langle \text{Mary will laugh}, s \rangle$

An example derivation

- 1 MERGE($\langle \text{laugh}, =d \ v \rangle$, $\langle \text{Mary}, d \ -k \rangle$)
- 2 MERGE($\langle \text{will}, =v \ +k \ s \rangle$, 1)
- 3 MOVE(2)

$\langle \text{laugh}, v \rangle$, $\langle \text{Mary}, -k \rangle$ $\langle \text{will laugh}, +k \ s \rangle$, $\langle \text{Mary}, -k \rangle$
 $\langle \text{Mary will laugh}, s \rangle$

Representing derivations

- The set of *possible derivations* in a grammar G is simply the set of terms over $\mathcal{F} \cup Lex$
 - ① each lexical item is a possible derivation (of itself)
 - ② given derivations t and t' , their merger is a possible derivation:
 $MERGE(t, t')$
 - ③ given a derivation t , applying the operation move to t is a possible derivation: $MOVE(t)$
- To characterize the complexity of the sound-meaning relation which is constitutive of a language in our sense, we need two things:
 - ① a characterization of the complexity of derivations (which we are for the moment taking as our meaning representations)
 - ② a characterization of the complexity of computing sounds from derivations

A Transductive Characterization of MGs

- Implicit in Michaelis [2001] is the fact that the **well-formed** MG derivations form a recognizable set of trees. (He shows that there is a fixed finite amount of information the operations MERGE and MOVE need to know to determine whether or not they apply; this gives us the states of our (bottom-up) tree automaton.)
- Deterministic finite-copying top-down tree transducers with regular look-ahead are able to map derivation trees into the derived objects they are the derivations of.

- We understand pretty well what kinds of languages we define, when we take minimalist grammars with the operations of `MOVE` (when it is constrained by the `SMC`) and `MERGE` as our theory of grammar.
- But there is more under heaven than is dreamt of in our theory!

- 1 The Basic Problem
- 2 Minimalist Grammars
- 3 Ellipsis**
- 4 Derivations with Delete

A statement of the problem

- As currently defined, minimalist grammars are able to pair up a great many sentences with appropriate meanings.
- However, there seems to be a systematic class of exceptions, where our grammars think that pieces are missing.
- We want to associate these exceptional sentences with appropriate meanings, and study the complexity of the resulting sound-meaning correspondences so defined.

John will kiss Mary, but Bill won't.

- The intuitive puzzle here is that, if we try to use our standard methods of computing meaning (assuming a structure like $S \rightarrow S \text{ Conj } S$), we end up with something like

$$\text{kiss}(m)(j) \ \& \ \llbracket \text{Bill won't} \rrbracket$$

- Somehow, from $\llbracket \text{Bill won't} \rrbracket$ we compute the meaning

$$\neg \text{kiss}(m)(b)$$

- ...and not the meanings

$$\neg \text{kiss}(j)(b)$$

$$\neg \text{praise}(m)(b)$$

Theories of Ellipsis

Ellipsis as an empty category

Here the syntactic structure of an elliptical sentence is as it appears:

Bill won't e

Where e is a non-pronounced element that acts like a pronoun.

Ellipsis as deletion

Here the syntactic structure of an elliptical sentence is much more complicated than it appears:

Bill won't ~~kiss Mary~~

On the other hand, the meaning of such sentences is computed as normal.

Ellipsis as LF-Copying

Bill won't e \rightarrow Bill won't kiss Mary

Theories of Ellipsis

Ellipsis as an empty category

Here the syntactic structure of an elliptical sentence is as it appears:

Bill won't e

Where e is a non-pronounced element that acts like a pronoun.

Ellipsis as deletion

Here the syntactic structure of an elliptical sentence is much more complicated than it appears:

Bill won't ~~kiss Mary~~

On the other hand, the meaning of such sentences is computed as normal.

Ellipsis as LF-Copying

Bill won't e \longrightarrow Bill won't kiss Mary

Theories of Ellipsis

Ellipsis as an empty category

Here the syntactic structure of an elliptical sentence is as it appears:

Bill won't e

Where e is a non-pronounced element that acts like a pronoun.

Ellipsis as deletion

Here the syntactic structure of an elliptical sentence is much more complicated than it appears:


Bill won't ~~kiss Mary~~

On the other hand, the meaning of such sentences is computed as normal.

Ellipsis as LF-Copying

Bill won't e \longrightarrow Bill won't kiss Mary


Deletion

- Deletion is the preferred method of describing ellipsis in the minimalist community.
 - John will kiss Mary, but Bill won't ~~kiss Mary~~.

- Under a deletion analysis, restrictions on what elliptical sentences mean are implemented by imposing constraints on what can be deleted.
 - 1 John will kiss Mary but Bill won't ~~kiss Mary~~.
 - 2 *John will kiss Mary but Bill won't ~~eat pasta regularly~~.
- This has its traditional formulation as “deletion up to recoverability.” More precise explications of this intuition recast ‘recoverability’ in terms of the existence of an appropriate antecedent.

Deletion up to recoverability

A structure t may be deleted only if there is a t' such that

- 1 t' is not deleted
- 2 t and t' are identical

- Deletion is the preferred method of describing ellipsis in the minimalist community.
 - John will kiss Mary, but Bill won't ~~kiss Mary~~.


identity
- Under a deletion analysis, restrictions on what elliptical sentences mean are implemented by imposing constraints on what can be deleted.
 - 1 John will kiss Mary but Bill won't ~~kiss Mary~~.
 - 2 *John will kiss Mary but Bill won't ~~eat pasta regularly~~.
- This has its traditional formulation as “deletion up to recoverability.” More precise explications of this intuition recast ‘recoverability’ in terms of the existence of an appropriate antecedent.

Deletion up to recoverability

A structure t may be deleted only if there is a t' such that

- 1 t' is not deleted
- 2 t and t' are identical

Three notions of identity

- A) Derivational identity
 - B) Derived tree identity
 - C) Semantic identity
 - identity of representation
 - semantic equivalence (identity of denotation)
-
- Derivational identity is arguably the most natural...
 - 1 the derivation is the structure computed by the parser
 - 2 items in a chart are derivational constituents...
 - 3 to compute the meaning/surface structure of an expression, we need first its derivation
 - It is easiest to work with.

$$\text{spellOut}(\text{DELETE}(t)) = \epsilon$$

$$\llbracket \text{DELETE}(t) \rrbracket = \llbracket t \rrbracket$$

Three notions of identity

- A) Derivational identity
- B) Derived tree identity
- C) Semantic identity
 - identity of representation
 - semantic equivalence (identity of denotation)
- Derivational identity is arguably the most natural...
 - 1 the derivation is the structure computed by the parser
 - 2 items in a chart are derivational constituents...
 - 3 to compute the meaning/surface structure of an expression, we need first its derivation
- It is easiest to work with.

$$\text{spellOut}(\text{DELETE}(t)) = \epsilon$$

$$\llbracket \text{DELETE}(t) \rrbracket = \llbracket t \rrbracket$$

Implementing Deletion Under Identity

- What happens once we enrich our stock of operations to include deletion, by adding the following case to our definition of possible derivations:
 - given a derivation t , applying the operation delete to t is a possible derivation: $\text{DELETE}(t)$
- and the following restriction on well-formed derivation trees:
 - If, in a derivation $d \in T_{\Sigma}$, there is a subpart $\text{DELETE}(t)$, then
 - 1 there must be another occurrence of t in d ,
 - 2 which is not deleted (there is no node labelled 'DELETE' on the path from the root of t to the root of d)

- How complex are derivation trees with this constraint on nodes labelled `DELETE`?
 - this condition is conceptually easy to state:

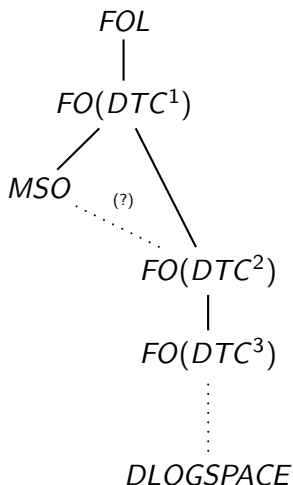
DELID

$\forall x$, if the parent of x is labeled `DELETE`, then $\exists y$ such that

- 1 **isomorphic**(x, y), and
- 2 the parent of y is *not* labeled `DELETE`

- however, **isomorphic** cannot be expressed in MSO, and I do not know what properties MSO extended with this predicate has
- Tiede and Kepser [2006] show that **isomorphic**(\cdot, \cdot) can be defined in $FO(DTC^2)$.

- The following is known [Engelfriet and Hoogeboom, 2007]:



- Because we do not know whether MSO is included in $FO(DTC^2)$, we do not know whether MG derivation tree languages are definable in $FO(DTC^2)$.
- I will show that MG derivation trees are in fact definable in $FO(DTC^1)$
- And that, where Φ_G is the $FO(DTC^1)$ sentence defining the well-formed derivation trees of G , $\Phi_G \wedge \text{DELID}$ is the $FO(DTC^2)$ sentence defining the well-formed derivation trees of G^{delete} .

- 1 The Basic Problem
- 2 Minimalist Grammars
- 3 Ellipsis
- 4 Derivations with Delete**

- *FOL* is extended to *FO(DTC¹)* by adding an operator *DTC* to the logic:
 - for ϕ a sentence, and x, y distinct variables free in ϕ , the following is a binary relation:

$$\text{DTC}_{x,y}\phi$$

- $\llbracket \text{DTC}_{x,y}\phi \rrbracket^g$ is the smallest reflexive and transitive relation R such that for every a and b , aRb if b is the unique element c such that

$$\llbracket \phi \rrbracket^{g[x:=a,y:=c]} = \mathbf{true}$$

- I assume a logic with a signature

$$\sigma = \langle (\ell^{(1)})_{\ell \in Lex}, \mathbf{r}^{(1)}, \mathbf{v}^{(1)}, \mathbf{d}, S_1^{(2)}, S_2^{(2)} \rangle$$

Where S_1 and S_2 are the binary relations interpreted as 'left-daughter' and 'right-daughter'

- We can define the following predicates:

| | |
|---|---|
| $x \triangleleft y$ | $S_1(x, y) \vee S_2(x, y)$ |
| $x \triangleleft^* y$ | $DTC_{y,x}[x \triangleleft y](y, x)$ |
| $x \triangleleft^+ y$ | $x \triangleleft^* y \wedge \neg x = y$ |
| leaf (x) | $\neg \exists y. x \triangleleft y$ |
| binary (x) | $\exists y, z. S_1(x, y) \wedge S_2(x, z)$ |
| unary (x) | $\neg(\mathbf{leaf}(x) \vee \mathbf{binary}(x))$ |
| left-dom [*] (x, y) | $DTC_{x,y}[S_1(x, y)](x, y)$ |
| head (x, y) | $\mathbf{leaf}(y) \wedge \mathbf{left-dom}^*(x, y)$ |

- I assume a logic with a signature

$$\sigma = \langle (\ell^{(1)})_{\ell \in Lex}, \mathbf{r}^{(1)}, \mathbf{v}^{(1)}, \mathbf{d}, S_1^{(2)}, S_2^{(2)} \rangle$$

Where S_1 and S_2 are the binary relations interpreted as 'left-daughter' and 'right-daughter'

- We can define the following predicates:

$$\frac{x \triangleleft^* y}{DTC_{y,x}[x \triangleleft y](y, x)}$$

A $FO(DTC^1)$ characterization of MG derivations

- Every useful lexical item has the following shape:

$$\langle v, \beta c \gamma \rangle$$

where

precat: β is a (possibly empty) sequence of $+x$ and $=x$ features

postcat: γ is a (possibly empty) sequence of $-x$ features

- I call a node in a model of a MG derivation **proper** iff
 - it is **binary**, and is labeled only **MERGE**
 - it is **unary**, and is labeled either only **MOVE** or only **DELETE**
 - it is a **leaf**, is labeled with exactly one ℓ , and
 - its **precat** features are satisfied,
 - its **postcat** features are satisfied,
 - and either
 - its **cat** feature is satisfied,
 - or it is the **head** of the **root**

MG derivation trees are models of...

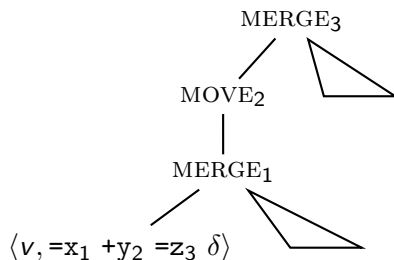
- Essentially, a node is proper iff it is a leaf and all of its features have been checked, or it is an internal node, appropriately labeled to allow correct feature checking.
- Given a MG G , its complete derivation trees (i.e. the set of derivation trees which evaluate to an expression whose only unchecked feature is a single categorial feature f at the (leaf which projects the) root) are models of the sentence

$$\Phi_G := \forall x. \mathbf{proper}(x)$$

- Extending G with deletion under derivational identity, its complete derivation trees are models of the sentence

$$\Phi_G \wedge \text{DELID}$$

- The first features in a lexical item ℓ are of the form $=x$ and $+x$ —with a $=x$ feature, ℓ selects an expression with categorial feature x , and with a $+x$ feature, ℓ triggers movement of an expression (contained in an expression) it has selected.



- at a leaf a , with label $\langle v, \beta c \gamma \rangle$, if the i^{th} feature in β is $=x$, there are two things that need to be done:

$$=f_i(a) \rightarrow$$

- verify that the i^{th} ancestor (y) of a is labeled 'MERGE', and that a is the head of this node (i.e. on a left-most path from this node)

$$\exists y. \mathbf{i-up}(a, y) \wedge \mathbf{r}(y) \wedge \dots$$

- verify that the head (w) of the right child (z) of y has as j^{th} feature a matching x , where j is the length of the path from this node to its right child

$$\exists z, w. S_2(y, z) \wedge \mathbf{leaf}(w) \wedge \left(\bigvee_j (\mathbf{j-up}(w, z) \wedge \mathbf{match}_j^i(a, w)) \right)$$

- at a leaf a , with label $\langle v, \beta c \gamma \rangle$, if the i^{th} feature in β is $+x$, there are three things that need to be done:

- verify that the i^{th} ancestor (y) of a is labeled 'MOVE', and that a is the head of y .

$$+f_i(a) \rightarrow \exists y. \mathbf{i-up}(a, y) \wedge \mathbf{v}(y) \wedge \dots$$

- verify that there is at least one leaf (z) beneath y which is distinct from a , and

$$\exists z. y \triangleleft^+ z \wedge \mathbf{leaf}(z) \wedge \neg x = z \wedge \dots$$

which checks one of its features at y , a matching $-x$

$$\left(\bigvee_j (\mathbf{check}_j(z, y) \wedge \mathbf{match}_j^i(a, z) \wedge \dots \right)$$

- verify that there is at most one such leaf (satisfying thereby the SMC)

$$\forall w. \left(\bigvee_j \mathbf{check}_j(w, y) \right) \rightarrow w = z$$

Definitions of derived predicate and relation symbols

$+f_i(y)$: The i^{th} feature of the label of y is a $+x$ feature, for some feature name x

$$\bigvee \ell(y) \quad - \quad \ell_i \text{ is a licensor feature}$$

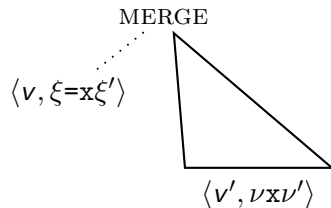
$\text{match}_j^i(y, z)$: The i^{th} feature of the label of y is the attractor version of the j^{th} feature of the label of z

$$\bigvee (\ell(x) \wedge \ell'(y)) \quad - \quad \ell_i \text{ is the attractor version of } \ell'_j$$

$\text{k-up}(y, z)$: Starting at y , if you go up k times (not counting nodes labeled DELETE), you get to z

$\text{checkable}_i(y, z)$: the leaf y 's i^{th} feature matches the feature the head (w) of z is checking at z

$$\exists w. \text{leaf}(w) \wedge \left(\bigvee_k \text{k-up}(w, z) \wedge \text{match}_i^k(w, y) \right)$$



- at a leaf a , with label $\langle v, \xi \rangle$, where c is the i^{th} feature in ξ , there are two things that need to be done:

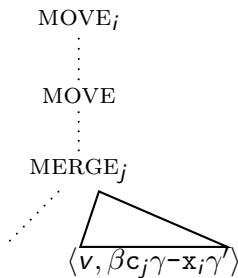
$$f_i(a) \rightarrow$$

- check that the last node y of which a is the head is i steps above a , and that it is the second argument to a node z labeled MERGE

$$\exists y. \mathbf{i-up}(x, y) \wedge \exists z. S_2(z, y) \wedge \mathbf{r}(z) \wedge \dots$$

- and that the head w of z checks a 's c feature at z

$$\exists w. \mathbf{leaf}(w) \wedge \left(\bigvee_j (\mathbf{j-up}(w, z) \wedge \mathbf{match}_i^j(w, x)) \right)$$



- at a leaf a , with label $\langle v, \beta c \gamma \rangle$, if the i^{th} feature of a is $-x$, there are two things that need to be done:

$$-f_i(a) \rightarrow \dots$$

- 1 there is a node (y) where a can check this feature

$$\exists y. \mathbf{check}_i(a, y) \wedge \dots$$

- 2 and a is the *only* expression which can check a feature at y

$$\forall z. (\bigvee_j \mathbf{check}_j(z, y)) \rightarrow z = a$$

The definition of $check_i$

$check_i(x, y)$: the i^{th} feature of x is checked at y

- There are i nodes, the i^{th} being y , the first properly dominating x , and each being properly dominated by the next, ...

$$\exists z_1, \dots, z_i. z_i = y \wedge z_1 \triangleleft^+ x \wedge \left(\bigwedge_{1 < j \leq i} z_j \triangleleft^+ z_{j-1} \right) \wedge \dots$$

- if z_k is the node at which x checks its c feature, ...

$$\bigwedge_{1 \leq k \leq i} (f_k(x) \rightarrow (\exists w. \mathbf{(k-1)\text{-up}}(x, w) \wedge S_2(z_k, w))) \wedge \dots$$

- then every node z_j above z_k is labeled MOVE, and x 's j^{th} feature is checkable at z_k , and ...

$$\bigwedge_{k < j \leq i} (\mathbf{v}(z_j) \wedge \mathbf{checkable}_j(x, z_j)) \wedge \dots$$

- at no node between z_j and z_{j-1} is the j^{th} feature of x checkable

$$\forall w. \left(\bigwedge_{k < j \leq i} (z_j \triangleleft^+ w \wedge w \triangleleft^+ z_{j-1} \rightarrow \neg \mathbf{checkable}_j(x, w)) \right)$$

- This result is not parochial to the deletion analysis of ellipsis: As long as the meanings recoverable in ellipsis sites are well characterized as the meanings of things we have derived previously, then the form meaning map will be expressible these terms.
- Because we are using $FO(DTC^2)$ to define our derivation trees, we are not able to link our characterization of the complexity of the form-meaning mappings to the literature on bismorphisms.
- We are really only using $FO(DTC^2)$ for the **isomorphic**(\cdot, \cdot) predicate. Would it be an improvement to represent identity via structure sharing (and switch to MSO or $FO(DTC^1)$ over DAGs)? To argue this, it seems necessary to show that DAG to tree homomorphisms acting on regular (or $FO(DTC^1)$) DAG sets are weaker than tree homomorphisms acting on $FO(DTC^2)$ tree sets.
- What happens when we add **isomorphic**(\cdot, \cdot) to MSO?

More Questions

- currently, the $FO(DTC^2)$ sentence Φ_G defining the well-formed derivations of a MG with deletion under identity G depends on G in the following ways:
 - the maximum number i such that $=f_i, +f_i, f_i, -f_i$, **match** $_j^i$, **checkable** $_i$, and **check** $_i$ exist (this is the length of the longest $\ell \in Lex$)
 - the definitions of $=f_i, +f_i, f_i, -f_i$, and **match** $_j^i(\cdot, \cdot)$
- linguists have proposed restrictions on phrase structure that amount to placing a finite upper bound (of two) on the number of $+x$ and $=x$ features a lexical item ℓ can have.
- Furthermore, the kinds of analyses linguists propose effectively bound the number of $-x$ features (to around three).
- Bounding the size of possible lexical items reduces the dependence of Φ_G on G to simply
 - the definitions of $=f_i, +f_i, f_i, -f_i$, and **match** $_j^i(\cdot, \cdot)$which is the kind of information present in the dependency structures from which Stabler shows (k -valued) MGs are identifiable in the limit.

- J. Engelfriet and H. J. Hoogeboom. Automata with nested pebbles capture first-order logic with transitive closure. *Logical Methods in Computer Science*, 3(2-3):1–27, 2007.
- J. Michaelis. *On Formal Properties of Minimalist Grammars*. PhD thesis, Universität Potsdam, 2001.
- E. P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer-Verlag, Berlin, 1997.
- H.-J. Tiede and S. Kepser. Monadic second-order logic and transitive closure logics over trees. *Electronic Notes in Theoretical Computer Science*, 165:189–199, 2006.