# Minimalist Grammars

## Formalisme en Dependency Structures

Sjoerd Dost    Myrthe Tielman

Logical Methods in NLP
03/05/12

# Inhoud

# The Minimalist Program

- Chomsky, Universal Grammar

# The Minimalist Program

- Chomsky, Universal Grammar
- "effort to attribute as little as possible to UG while still accounting for the apparent diversity of human languages" *(Stabler, 2011)*

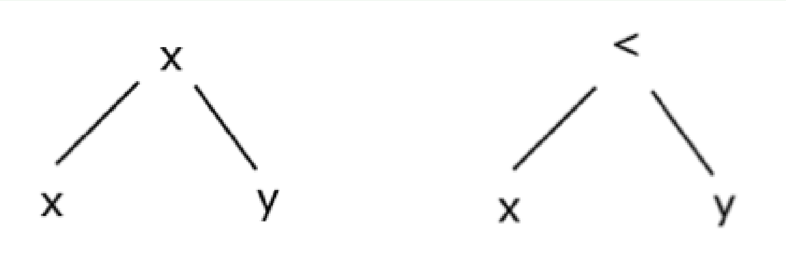
# The Minimalist Program

- Chomsky, Universal Grammar
- "effort to attribute as little as possible to UG while still accounting for the apparent diversity of human languages" *(Stabler, 2011)*
- Endocentricity

# Endocentricity

"A head X determines certain relevant properties of the phrase XP it is the head of."

Merging head x with xy gives {x{xy}}



# Lexicon

Stabler 2010: *Computational perspectives on minimalism*

A minimalist grammar G is a lexicon:

$$G \subset PhoneticFeatures \; x \; Features^{*}, a \; finite \; set$$

Where Features* is the set of finite sequences of syntactic features
Where the elements of the lexicon are combined by the merge operation. *(Stabler, 2011)*

# Features

Features:

- Categorial

$$N, V, A, P, C, T, D, \ldots$$

# Features

Features:

- Categorial

$$N, V, A, P, C, T, D, \ldots$$

- Selector

$$=N, =V, =A, =P, =C, =T, =D, \ldots$$

# Features

Features:

- Categorial

$$N, V, A, P, C, T, D, \ldots$$

- Selector

$$=N, =V, =A, =P, =C, =T, =D, \ldots$$

- Licensee (goal)

$$-wh, -case, \ldots$$

# Features

Features:

- Categorial

$$N, V, A, P, C, T, D, \ldots$$

- Selector

$$=N, =V, =A, =P, =C, =T, =D, \ldots$$

- Licensee (goal)

$$-wh, -case, \ldots$$

- Licensor (probes)

$$+wh, +case, \ldots$$

# Example Lexicon

**Who Marie praises?**

$$
\begin{aligned}
\text{Marie} &:: \text{D} \\
\text{who} &:: \text{D -wh} \\
\text{praises} &:: =\text{D} =\text{D V} \\
\varepsilon &:: =\text{T} +\text{wh C}
\end{aligned}
$$

# Merge

- From lexical items to trees

# Merge

- From lexical items to trees
- Lexical items are 1 node trees

# Merge

- From lexical items to trees
- Lexical items are 1 node trees
- Merge (external merge) and Move (internal merge)

# Merge

- From lexical items to trees
- Lexical items are 1 node trees
- Merge (external merge) and Move (internal merge)
- Merge to merge 2 trees together

# Merge

- From lexical items to trees
- Lexical items are 1 node trees
- Merge (external merge) and Move (internal merge)
- Merge to merge 2 trees together
- Move to restructure trees

# Merge rule

$$em(t_1[=x], t_2[x]) = \begin{cases} \underset{t_1 \quad t_2}{<} & \text{if } |t_1| = 1 \\ \underset{t_2 \quad t_1}{>} & \text{otherwise} \end{cases}$$

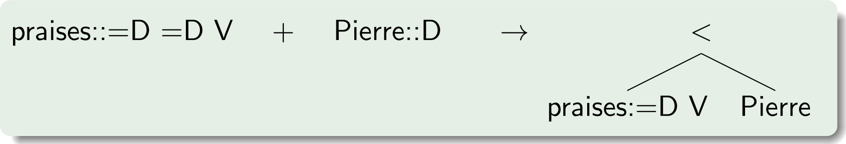
$$\frac{s :: =f\gamma \qquad t \cdot f, \alpha_1, \ldots \alpha_k}{st : \gamma, \alpha_1, \ldots, \alpha_k}\texttt{merge1}$$

$$\frac{s : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\texttt{merge2}$$

$$\frac{s \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\texttt{merge3}$$

# Merge rule

Merge (external merge)




# Move

Move (internal merge)

- SMC: Exactly one head in the tree has -x as its first feature.

# Move

Move (internal merge)

- SMC: Exactly one head in the tree has -x as its first feature.
- t$\{t_1 \mapsto t_2\}$ : the result of replacing subtree $t_1$ by $t_2$ in $t$.

# Move

Move (internal merge)

- SMC: Exactly one head in the tree has -x as its first feature.
- t$\{t_1 \mapsto t_2\}$ : the result of replacing subtree $t_1$ by $t_2$ in $t$.
- $t^{\mathrm{M}}$ : the maximal projection of the head of $t$.

# Move

Move (internal merge)

- SMC: Exactly one head in the tree has -x as its first feature.
- t$\{t_1 \mapsto t_2\}$ : the result of replacing subtree $t_1$ by $t_2$ in $t$.
- $t^{\mathrm{M}}$ : the maximal projection of the head of $t$.
- A subtree is a maximal projection, if it is not properly included in any larger subtree that has the same head.

# Move

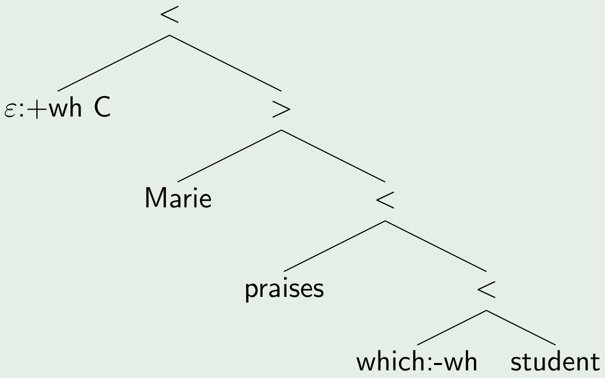
$$im(t_1[+x]) = \underset{\displaystyle t^{M}{}_{2} \qquad t_1\{t_2[-x]^{M} \mapsto \varepsilon\}}{>} \qquad \textit{if SMC}.$$

$$\frac{s : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \ldots, \alpha_k}{ts : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k}\,\mathtt{move1}$$

$$\frac{s : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\,\mathtt{move2}$$

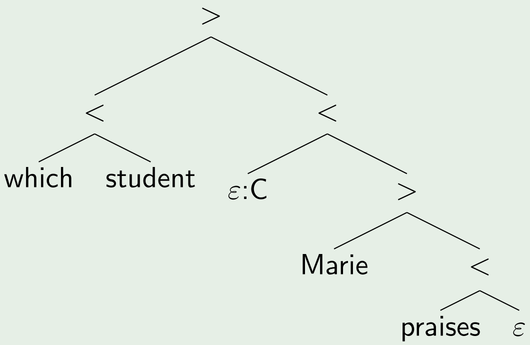
# Move

Move (internal merge)



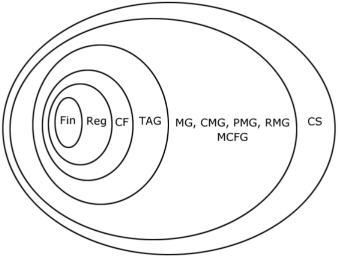
# Structuurregels

Move (internal merge)



# Onion

CMG = Conflated Minimalist Grammar, PMG = Phase-based Minimalist Grammar, RMG = Relativized Minimalist Grammar, MCFG = Multiple Context-free Grammar



"An interpretation of Minimalist Grammars in terms of dependency structures." *(Boston, Hale Kuhlmann 2010)*

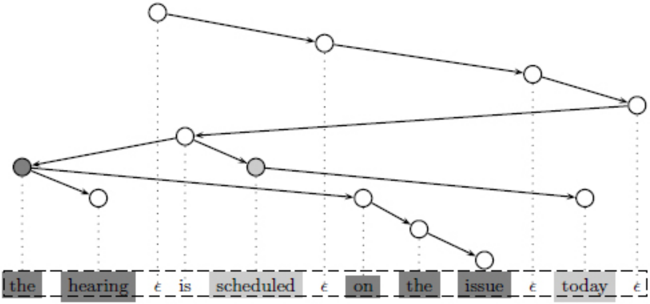


**Table 1.** Merge and Move

$$\frac{s :: =f\gamma \qquad t \cdot f, \alpha_1, \ldots \alpha_k}{st : \gamma, \alpha_1, \ldots, \alpha_k}\texttt{merge1}$$

$$\frac{s : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\texttt{merge2}$$

$$\frac{s \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\texttt{merge3}$$

$$\frac{s : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \ldots, \alpha_k}{ts : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move1}$$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

**Table 2.** Merge in terms of dependency trees

$$\frac{(\{\lambda\}, \langle\lambda\rangle) :: =f\gamma \qquad (\mathbf{t}, x) \cdot f, \alpha_1, \ldots, \alpha_k}{(\{\lambda\} \cup \uparrow_1 \mathbf{t}, \langle\lambda\rangle \cdot \uparrow_1 x) : \gamma, \uparrow_1 \alpha_1, \ldots, \uparrow_1 \alpha_k} \texttt{merge1}_{DG}$$

$$\frac{(\mathbf{s}, x) : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad (\mathbf{t}, y) \cdot f, \iota_1, \ldots, \iota_l}{(\mathbf{s} \cup \uparrow_i \mathbf{t}, \uparrow_i y \cdot x) : \gamma, \alpha_1, \ldots, \alpha_k, \uparrow_i \iota_1, \ldots, \uparrow_i \iota_l} \texttt{merge2}_{DG}$$

$$\frac{(\mathbf{s}, x) \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad (\mathbf{t}, y) \cdot f\delta, \iota_1, \ldots, \iota_l}{(\mathbf{s}, x) : \gamma, \alpha_1, \ldots, \alpha_k, (\uparrow_i \mathbf{t}, \uparrow_i y) : \delta, \uparrow_i \iota_1, \ldots, \uparrow_i \iota_l} \texttt{merge3}_{DG}$$

where $i = \mathit{next}((\mathbf{s}, x) \cdot =f\gamma, \alpha_1, \ldots, \alpha_k)$

**Table 3.** Move in terms of dependency trees

$$\frac{(\mathbf{s}, x) : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, (\mathbf{t}, y) : -f, \alpha_{i+1}, \ldots, \alpha_k}{(\mathbf{s} \cup \mathbf{t}, yx) : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k} \texttt{move1}_{DG}$$

$$\frac{\mathbf{s} \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{\mathbf{s} : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : \delta, \alpha_{i+1}, \ldots, \alpha_k} \texttt{move2}_{DG}$$

$$\frac{s :: =f\gamma \qquad t \cdot f, \alpha_1, \ldots \alpha_k}{st : \gamma, \alpha_1, \ldots, \alpha_k}\texttt{merge1}$$

$$\frac{s : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\texttt{merge2}$$

$$\frac{s \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\texttt{merge3}$$

$$\frac{s :: =f\gamma \qquad t \cdot f, \alpha_1, \ldots \alpha_k}{st : \gamma, \alpha_1, \ldots, \alpha_k}\texttt{merge1}$$

$$\frac{s : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\texttt{merge2}$$

$$\frac{s \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\texttt{merge3}$$

the::=n d    boat::n    docked::=d=d v    where::d -wh    $\epsilon$::=v +wh c

(a) Lexicon 1.

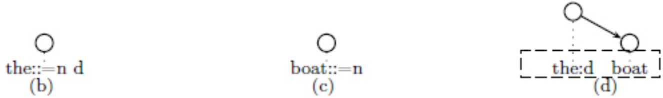



**Fig. 1.** **merge1**$_{DG}$ applies to two simple dependency trees

$$\frac{s : =f\gamma, \alpha_1, \dots, \alpha_k \qquad t \cdot f, \iota_1, \dots, \iota_l}{ts : \gamma, \alpha_1, \dots, \alpha_k, \iota_1, \dots, \iota_l}\,\texttt{merge2} \qquad \frac{s \cdot =f\gamma, \alpha_1, \dots, \alpha_k \qquad t \cdot f\delta, \iota_1, \dots, \iota_l}{s : \gamma, \alpha_1, \dots, \alpha_k, t : \delta, \iota_1, \dots, \iota_l}\,\texttt{merge3}$$

$$\frac{(\mathbf{s}, x) : =f\gamma, \alpha_1, \dots, \alpha_k \qquad (\mathbf{t}, y) \cdot f, \iota_1, \dots, \iota_l}{(\mathbf{s} \cup \uparrow_i \mathbf{t}, \uparrow_i y \cdot x) : \gamma, \alpha_1, \dots, \alpha_k, \uparrow_i \iota_1, \dots, \uparrow_i \iota_l}\,\texttt{merge2}_{DG} \qquad \frac{(\mathbf{s}, x) \cdot =f\gamma, \alpha_1, \dots, \alpha_k \qquad (\mathbf{t}, y) \cdot f\delta, \iota_1, \dots, \iota_l}{(\mathbf{s}, x) : \gamma, \alpha_1, \dots, \alpha_k, (\uparrow_i \mathbf{t}, \uparrow_i y) : \delta, \uparrow_i \iota_1, \dots, \uparrow_i \iota_l}\,\texttt{merge3}$$

$$\frac{s : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f, \iota_1, \ldots, \iota_l}{ts : \gamma, \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l}\,\texttt{merge2}$$

$$\frac{(\mathbf{s}, x) : =f\gamma, \alpha_1, \ldots, \alpha_k \qquad (\mathbf{t}, y) \cdot f, \iota_1, \ldots, \iota_l}{(\mathbf{s} \cup \uparrow_i \mathbf{t}, \uparrow_i y \cdot x) : \gamma, \alpha_1, \ldots, \alpha_k, \uparrow_i \iota_1, \ldots, \uparrow_i \iota_l}\,\texttt{merge2}_{DG}$$

$$\frac{s \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad t \cdot f\delta, \iota_1, \ldots, \iota_l}{s : \gamma, \alpha_1, \ldots, \alpha_k, t : \delta, \iota_1, \ldots, \iota_l}\,\texttt{merge3}$$

$$\frac{(\mathbf{s}, x) \cdot =f\gamma, \alpha_1, \ldots, \alpha_k \qquad (\mathbf{t}, y) \cdot f\delta, \iota_1, \ldots, \iota_l}{(\mathbf{s}, x) : \gamma, \alpha_1, \ldots, \alpha_k, (\uparrow_i \mathbf{t}, \uparrow_i y) : \delta, \uparrow_i \iota_1, \ldots, \uparrow_i \iota_l}\,\texttt{merge3}_{DG}$$

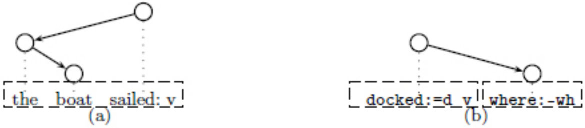



**Fig. 2.** $\texttt{merge2}_{DG}$ and $\texttt{merge3}_{DG}$

$$\frac{s : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \ldots, \alpha_k}{ts : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move1}$$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

$$\frac{s : +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \ldots, \alpha_k}{ts : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move1}$$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

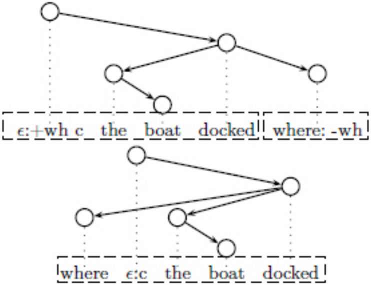



**Fig. 3.** move1$_{DG}$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

$$\frac{\mathbf{s} \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{\mathbf{s} : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}_{DG}$$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

$$\frac{\mathbf{s} \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{\mathbf{s} : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}_{DG}$$

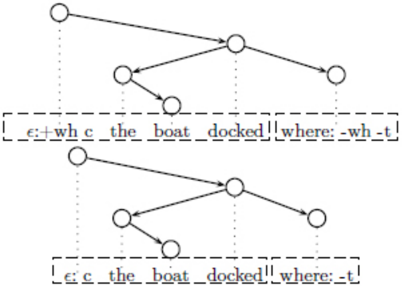



**Fig. 4.** move2$_{DG}$

$$\frac{s \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, t : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{s : \gamma, \alpha_1, \ldots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}$$

$$\frac{\mathbf{s} \cdot +f\gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : -f\delta, \alpha_{i+1}, \ldots, \alpha_k}{\mathbf{s} : \gamma, \alpha_1, \ldots, \alpha_{i-1}, \mathbf{t} : \delta, \alpha_{i+1}, \ldots, \alpha_k}\texttt{move2}_{DG}$$

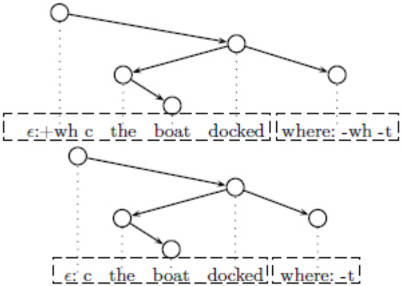



**Fig. 4.** $\texttt{move2}_{DG}$

(SMC) None of $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k$ has -f as its first feature

- Projectivity: subtrees span intervals

- Projectivity: subtrees span intervals
- Non-projective structures violate this constraint

- Projectivity: subtrees span intervals
- Non-projective structures violate this constraint
- Block degree: maximum number of intervals spanned

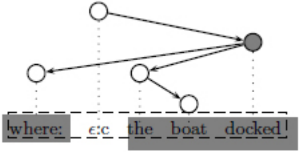



**Fig. 5.** The block degree of this structure is 2

- Merge always forms dependency relations between roots of subtrees by construction

- Merge always forms dependency relations between roots of subtrees by construction
- Move1 can create non-projective structures

- Merge always forms dependency relations between roots of subtrees by construction
- Move1 can create non-projective structures
- Movement is triggered by lincensor-licensee pairs: block degrees bounded by #licensees

- Wellnested structures prohibit thecrossing of disjoint subtree intervals.

- Wellnested structures prohibit thecrossing of disjoint subtree intervals.
- Not all mildly context-sensitive formalisms can derive illnested structures (i.e. TAG).

MGs can derive illnested structures (see example)