Efficient Third-Order Dependency Parsers

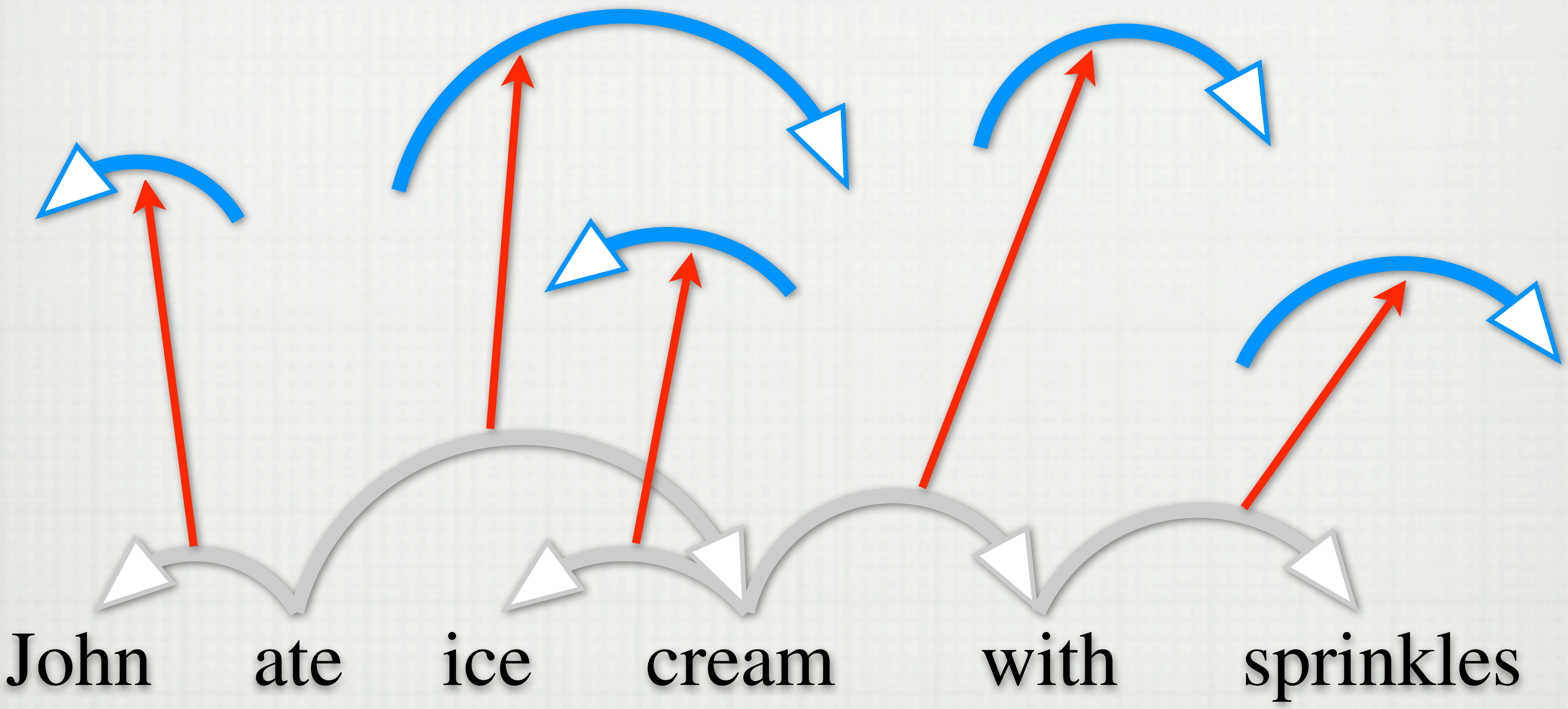
Terry Koo and Michael Collins
MIT CSAIL

Dependency Parsing

- Syntax represented by head-modifier dependencies
- Parsing is a search for the highest-scoring tree

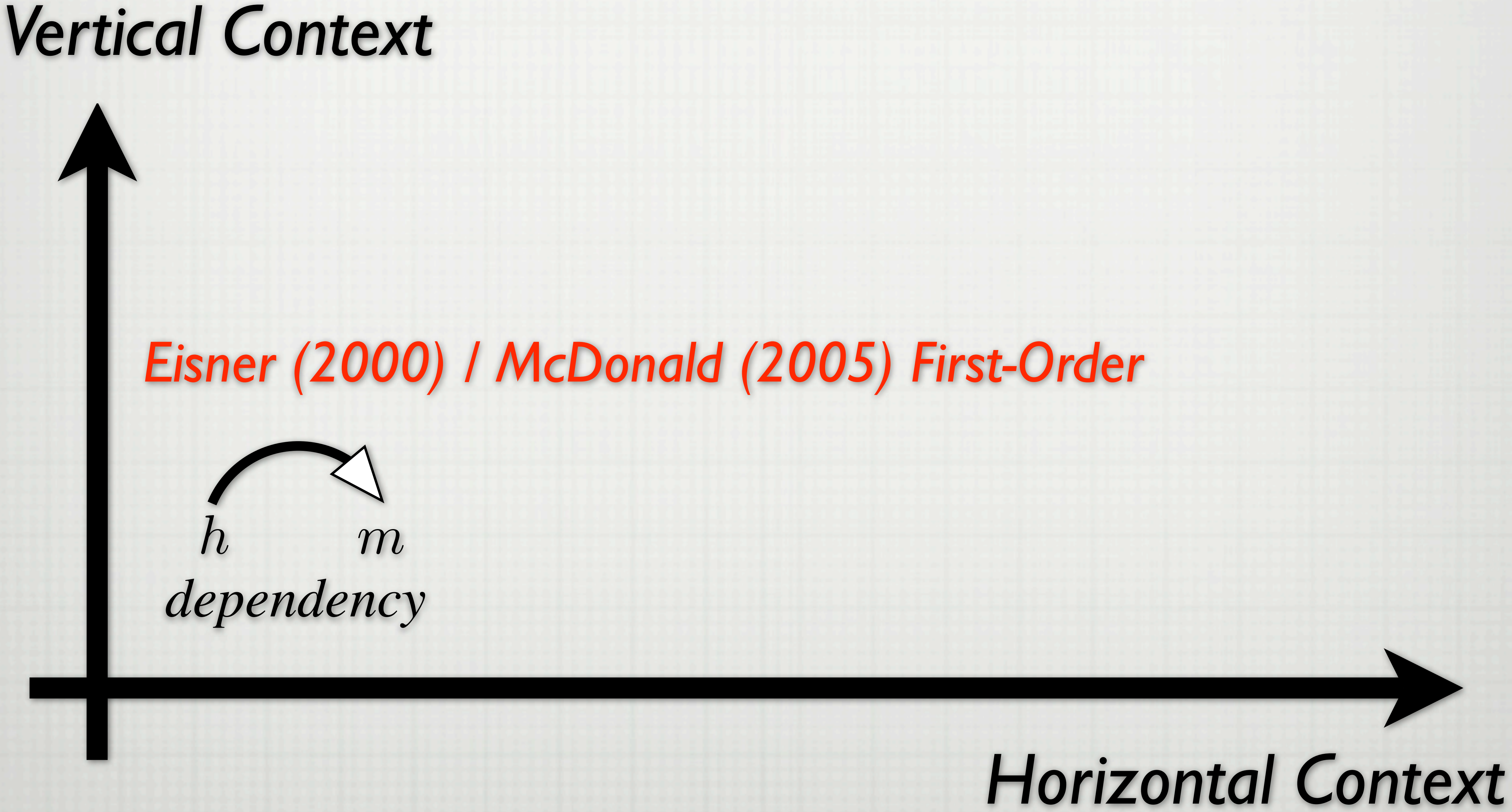
$$y^{\star}(\mathbf{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}(\mathbf{x})} \textsc{Score}(\mathbf{x}, y)$$

Factored (Graph-based) Parsing

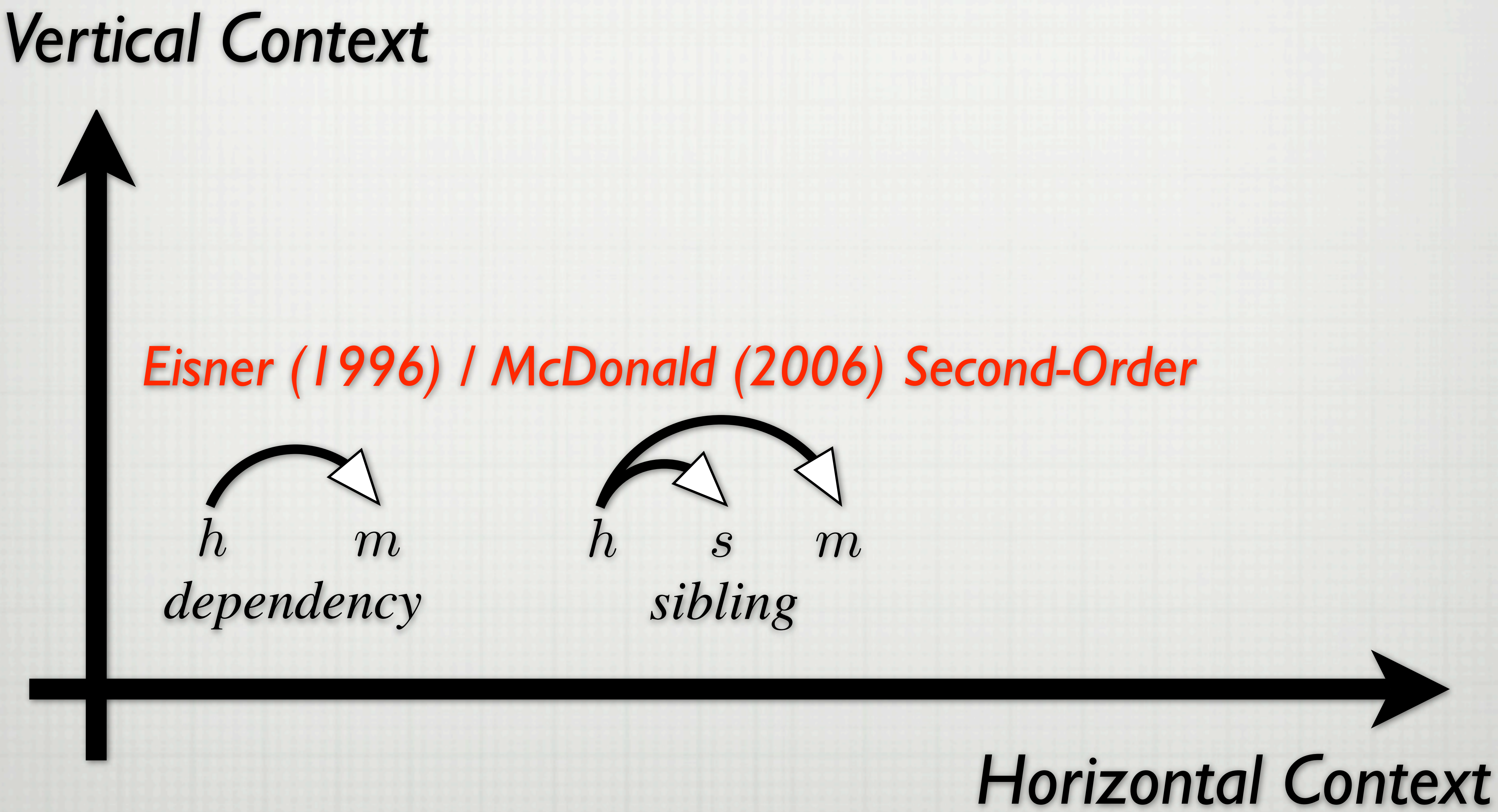


- Decompose $\textsc{Score}(\mathbf{x}, y) = \sum_{p \in y} \textsc{Score}(\mathbf{x}, p)$

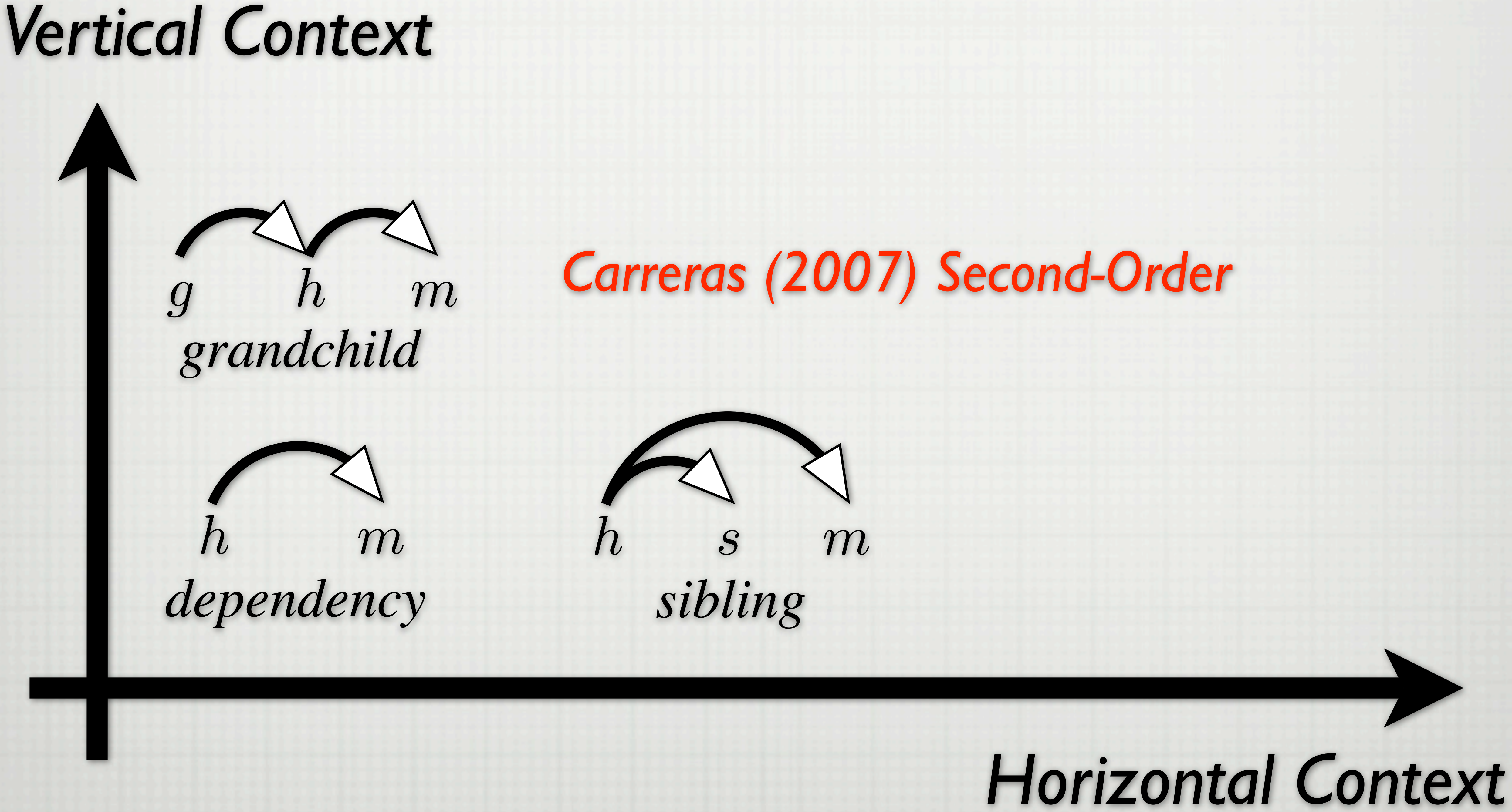
Higher-Order Factorizations



Higher-Order Factorizations

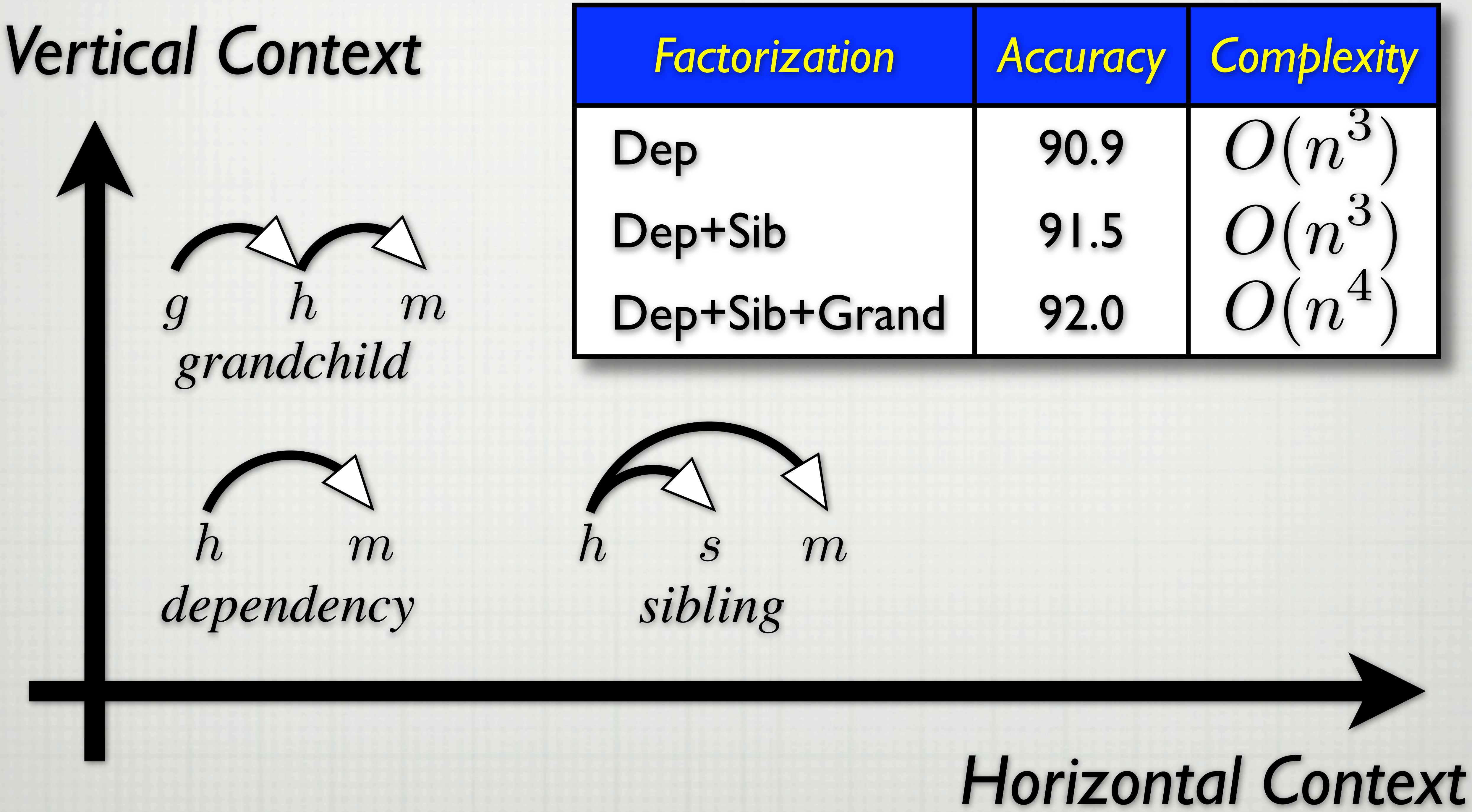


Higher-Order Factorizations

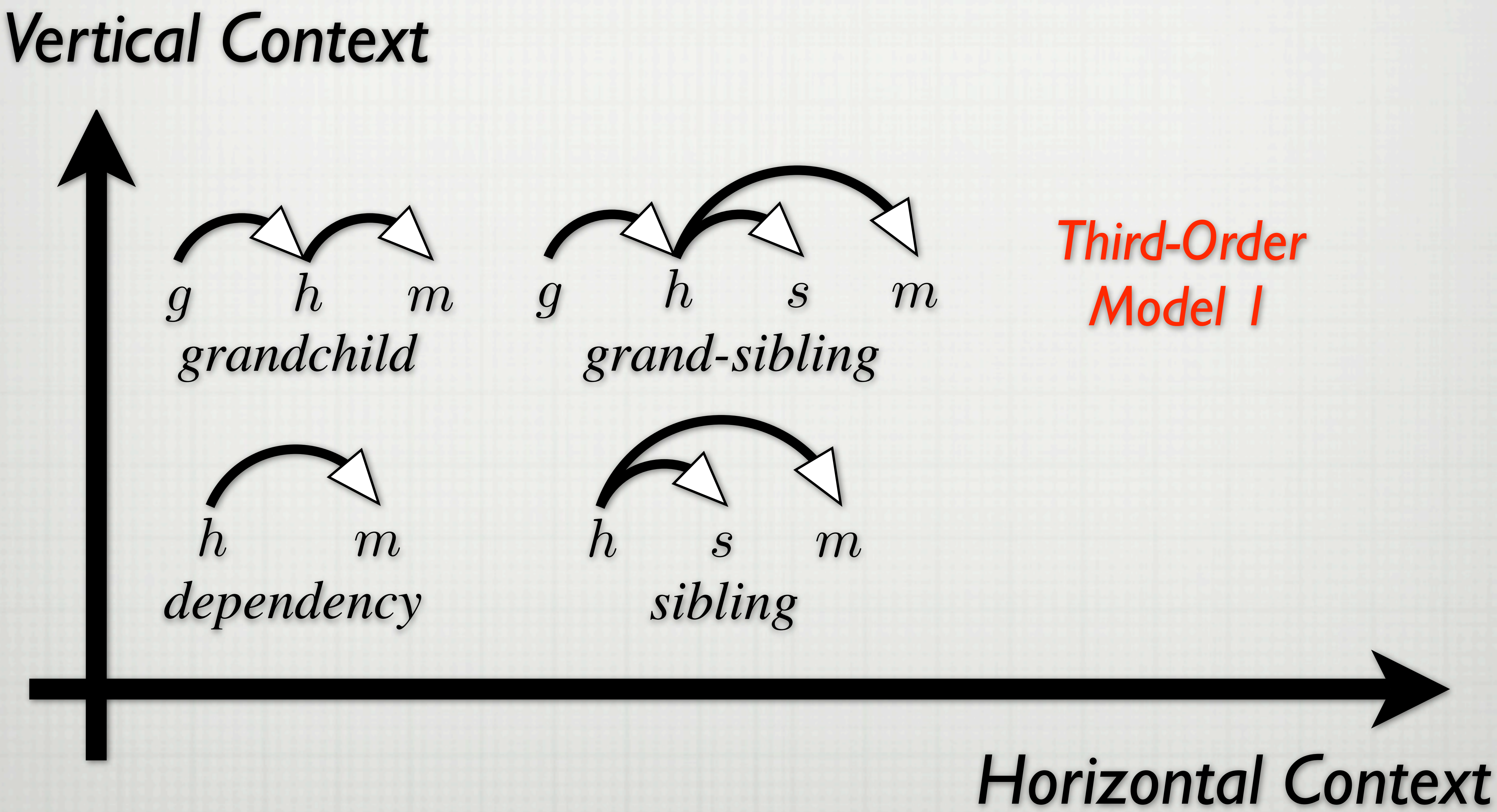


Higher-Order Factorizations

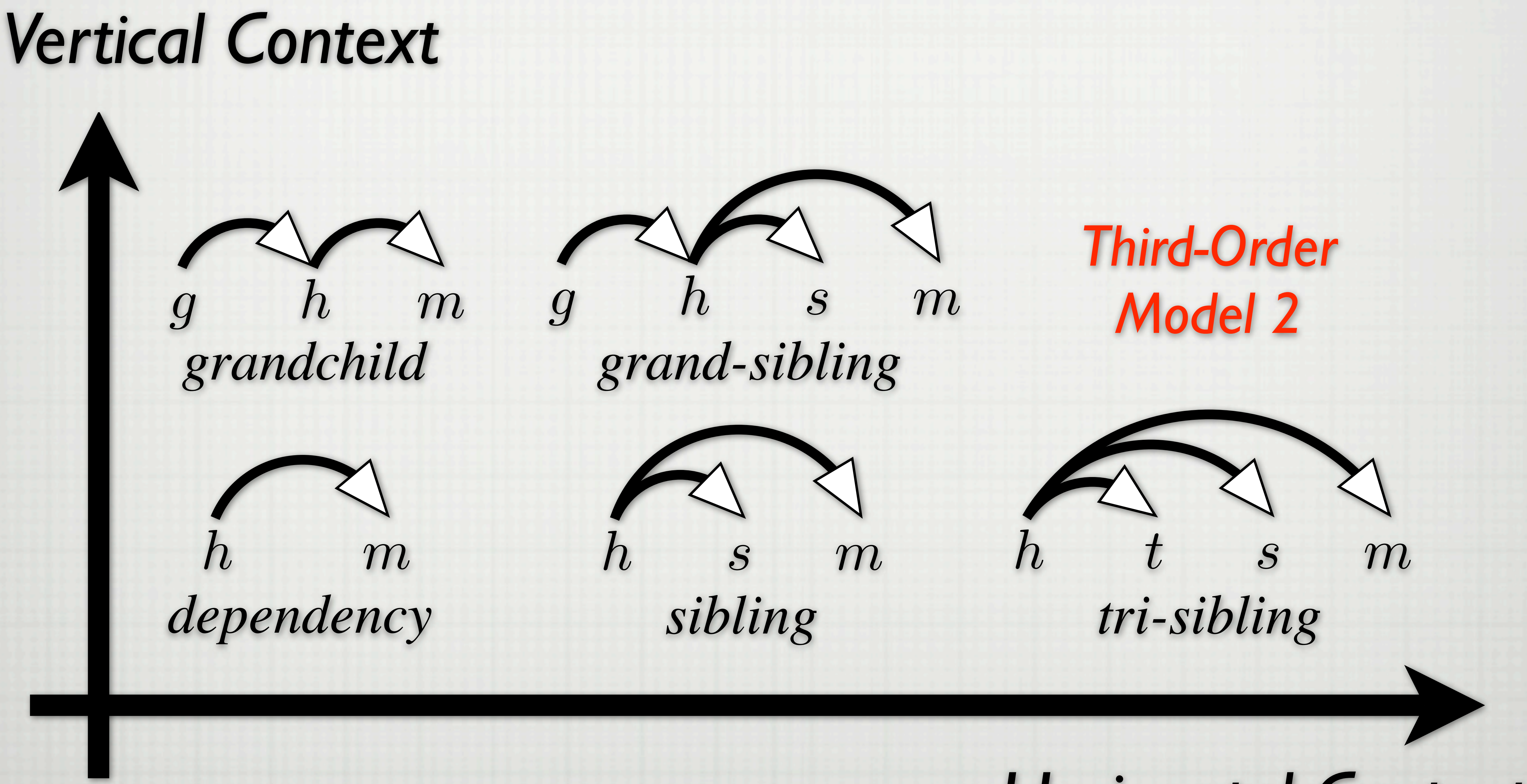
Factorization	*Accuracy*	*Complexity*
Dep	90.9	$O(n^3)$
Dep+Sib	91.5	$O(n^3)$
Dep+Sib+Grand	92.0	$O(n^4)$



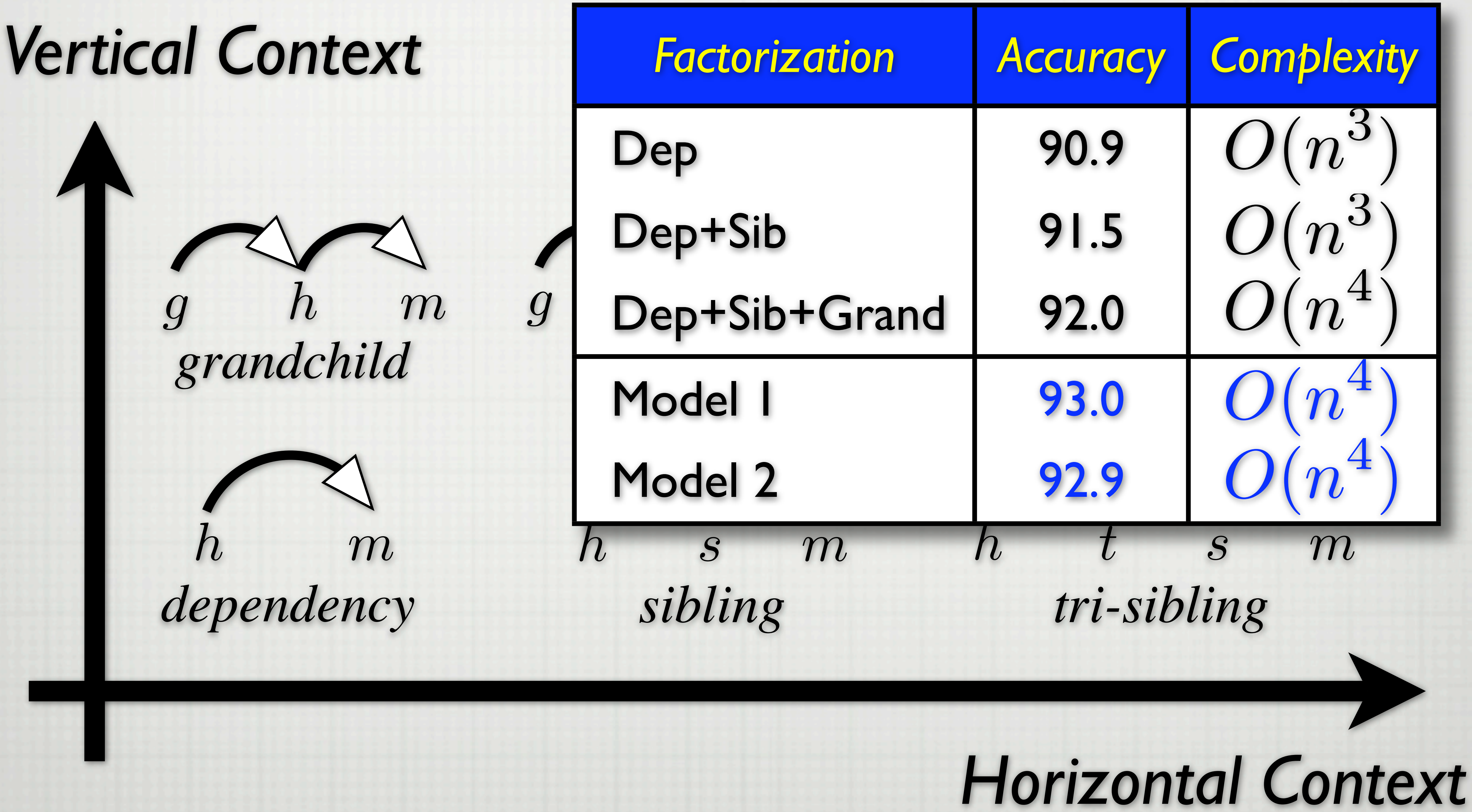
Higher-Order Factorizations



Higher-Order Factorizations



Higher-Order Factorizations

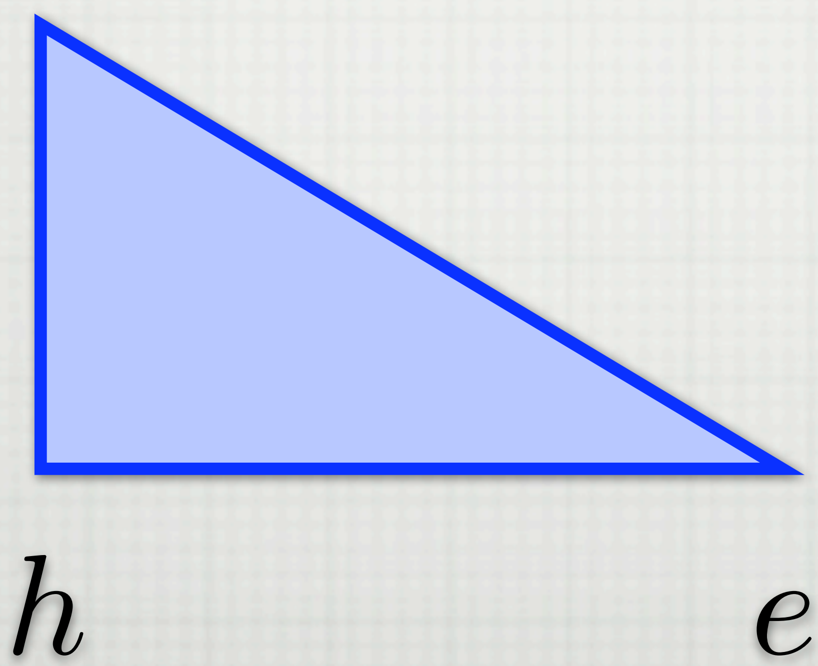


Factorization	Accuracy	Complexity
Dep	90.9	$O(n^3)$
Dep+Sib	91.5	$O(n^3)$
Dep+Sib+Grand	92.0	$O(n^4)$
Model 1	93.0	$O(n^4)$
Model 2	92.9	$O(n^4)$

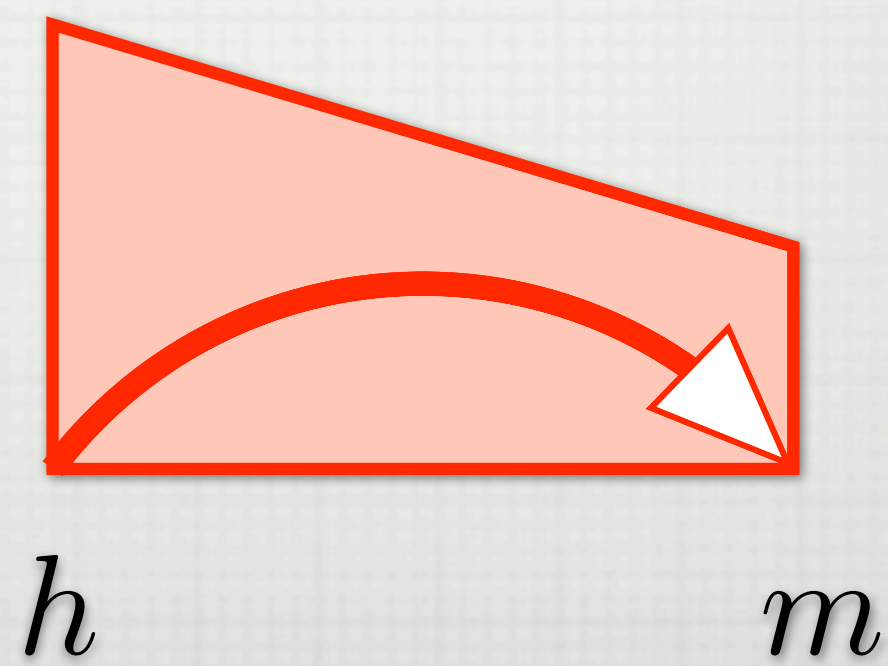
First-Order Parser

- Eisner (2000) algorithm: $O(n^3)$

Complete Span
A "half-constituent"

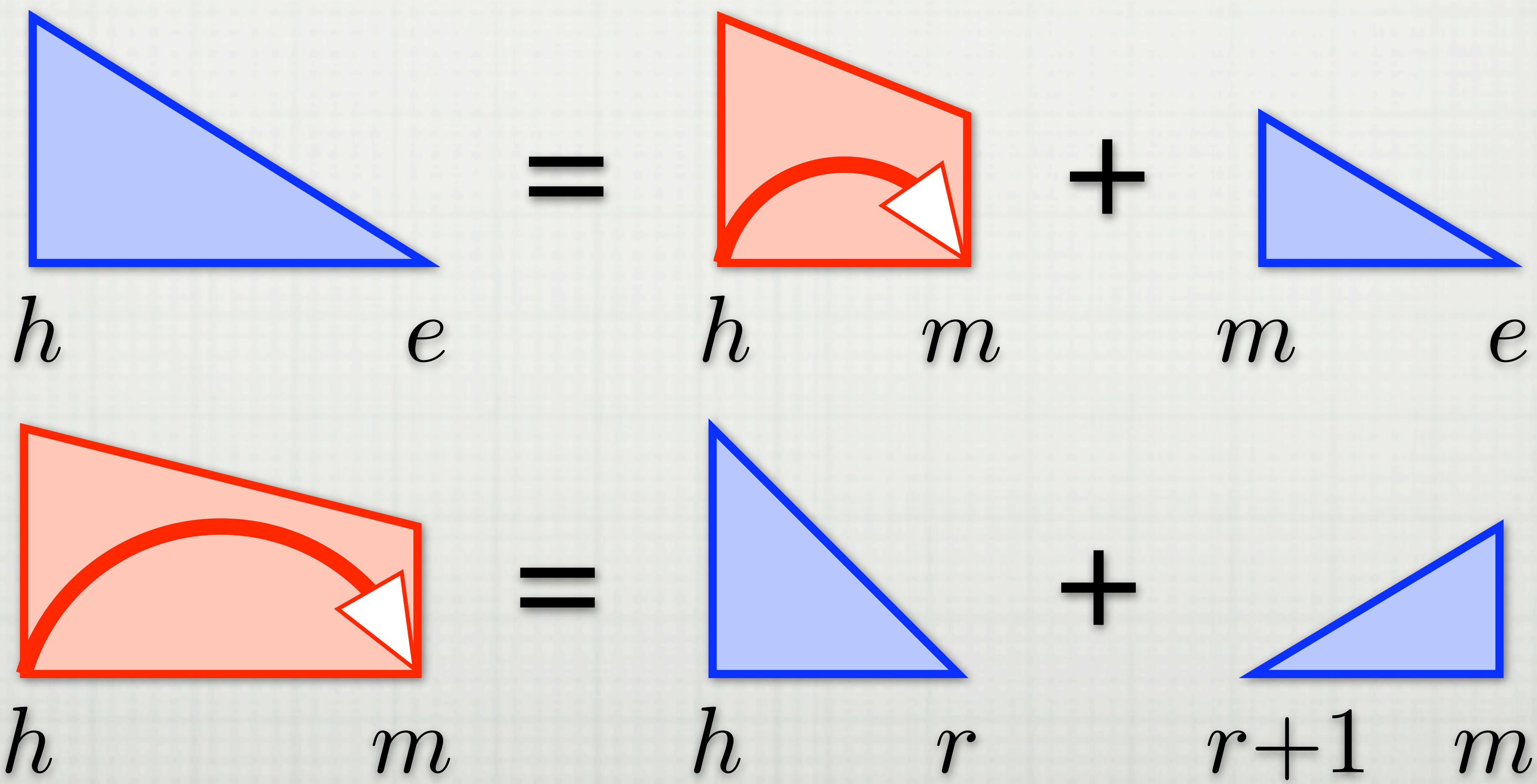


Incomplete Span
A dependency



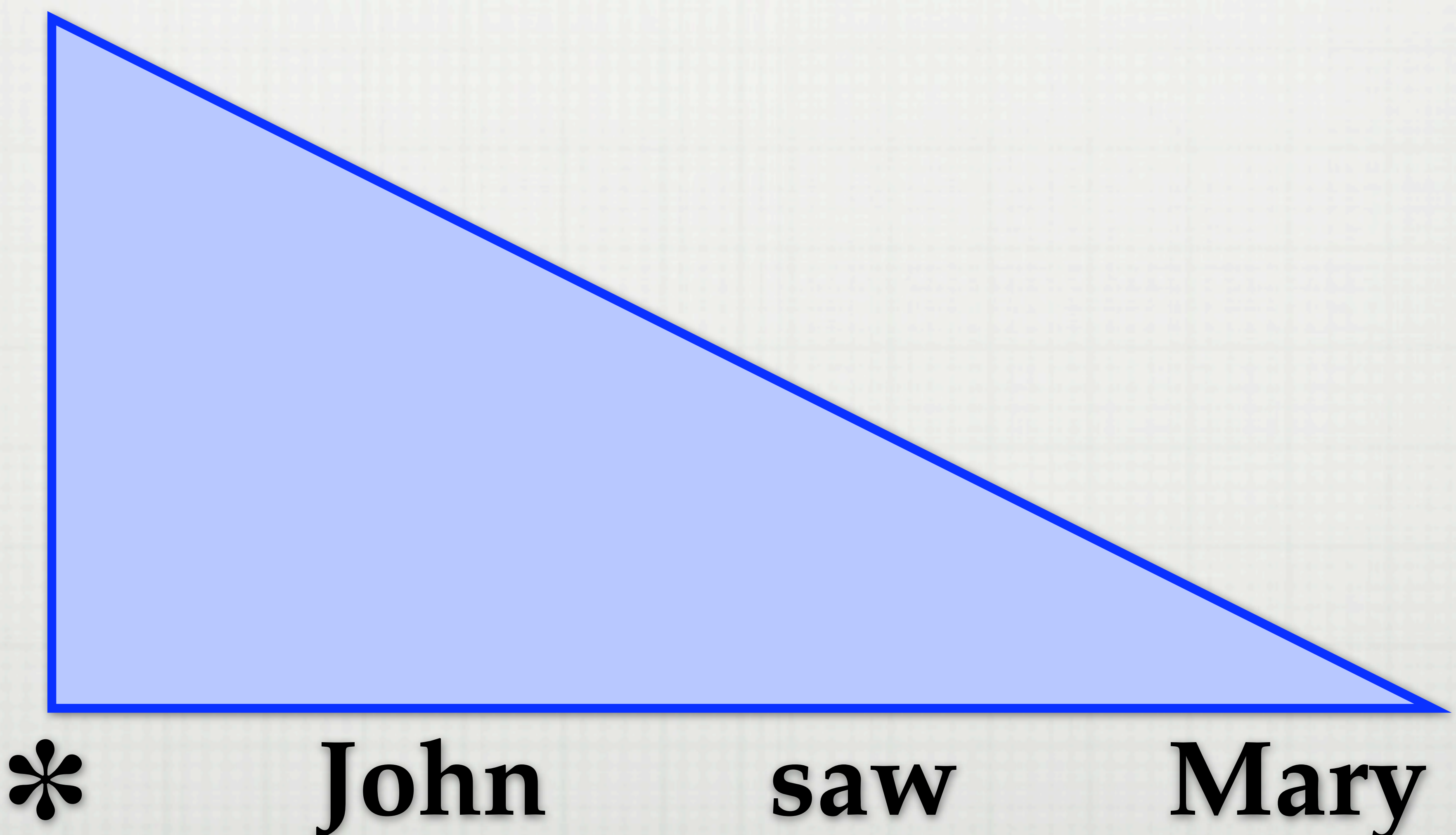
First-Order Parser

- Eisner (2000) algorithm: $O(n^3)$
  - Derivation of *complete* and *incomplete spans*:



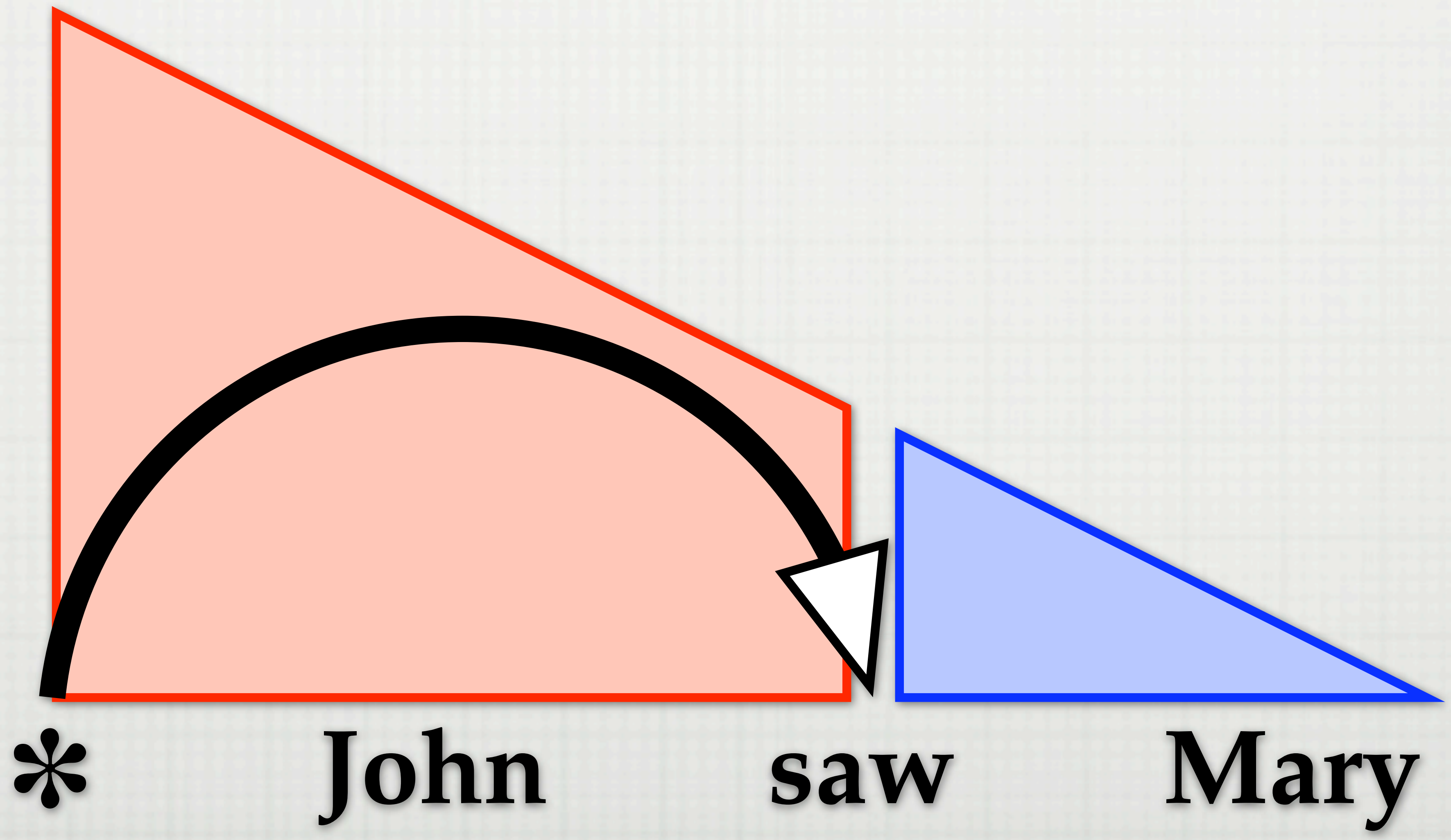
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$



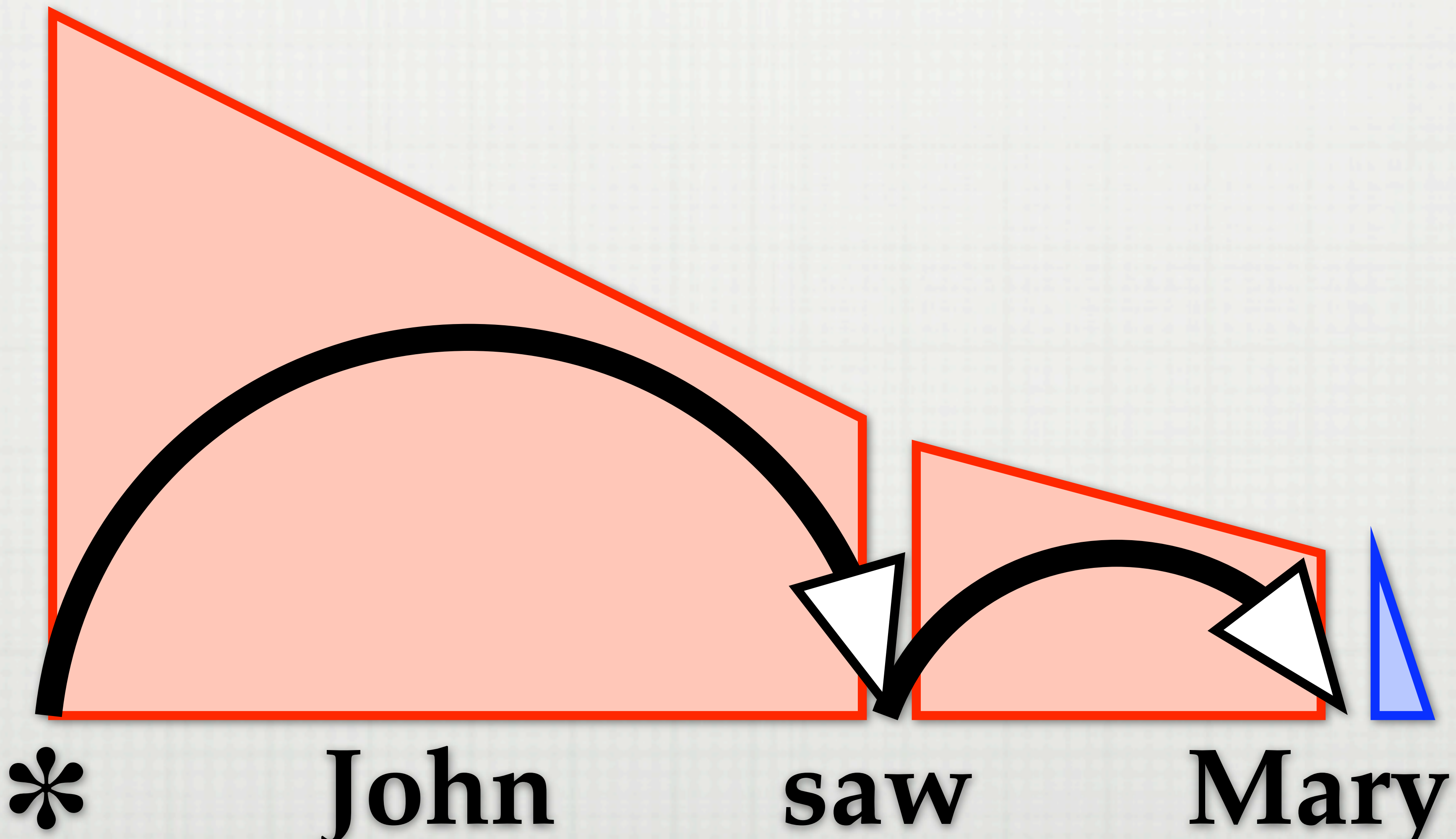
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$



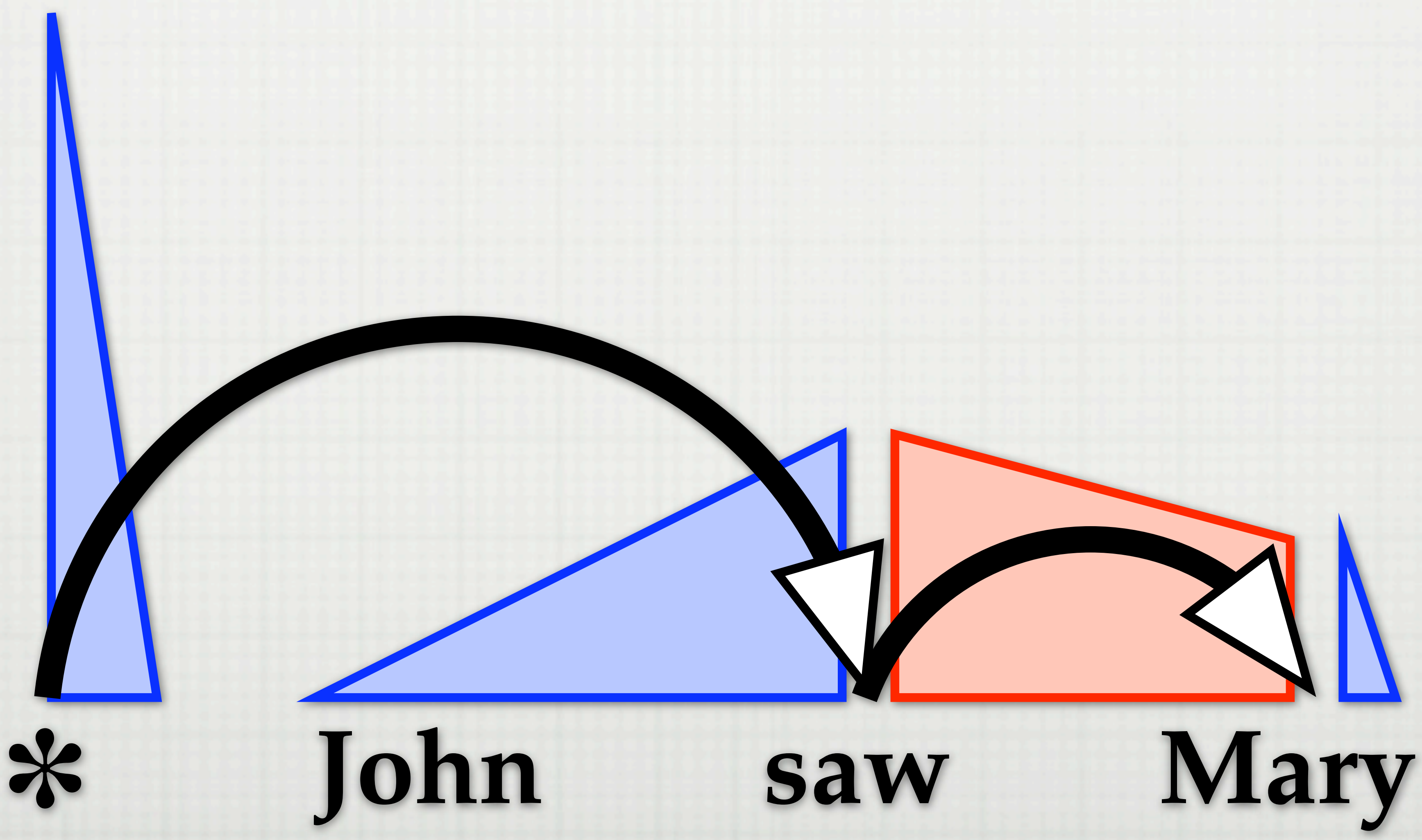
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$



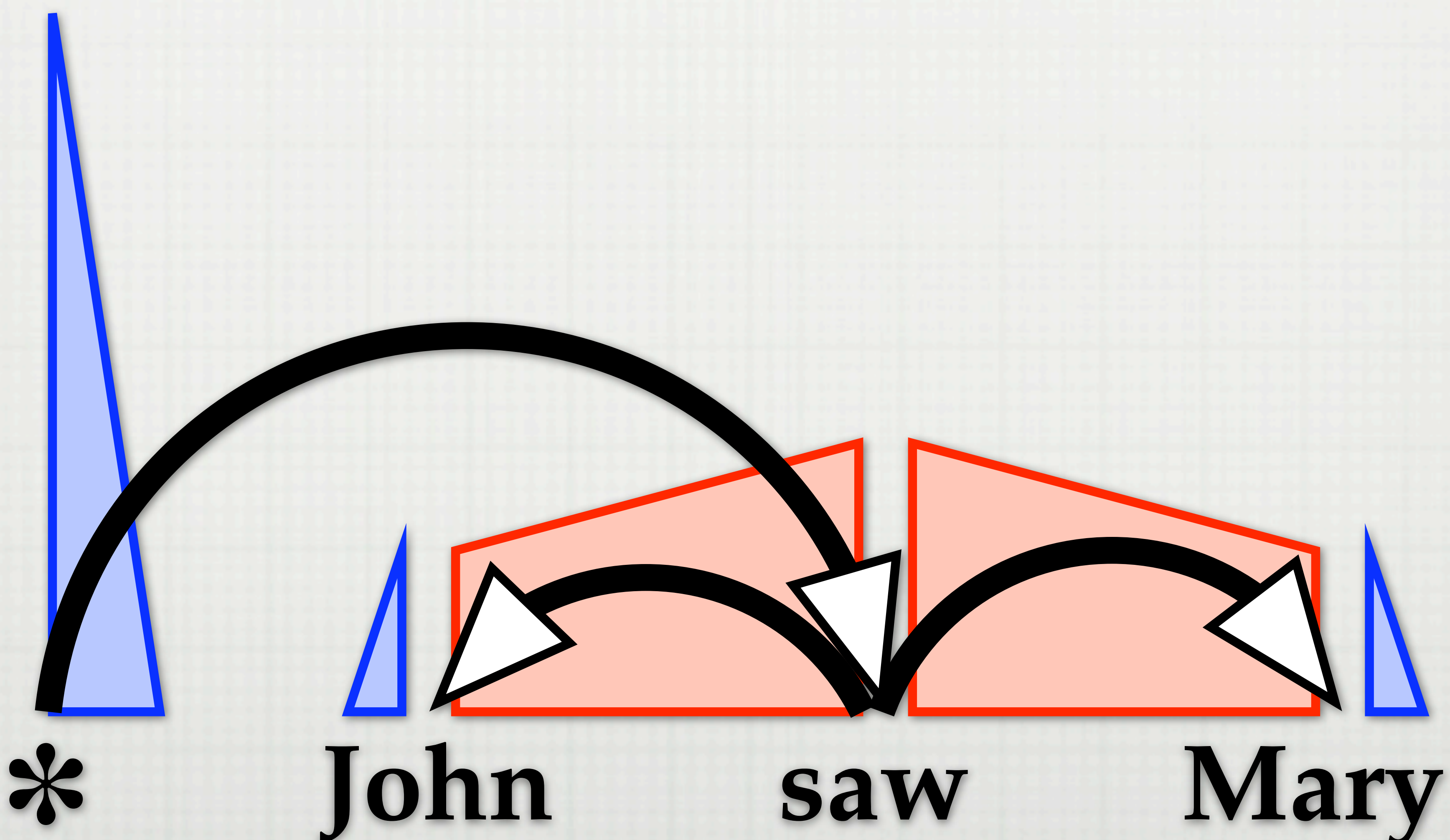
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$



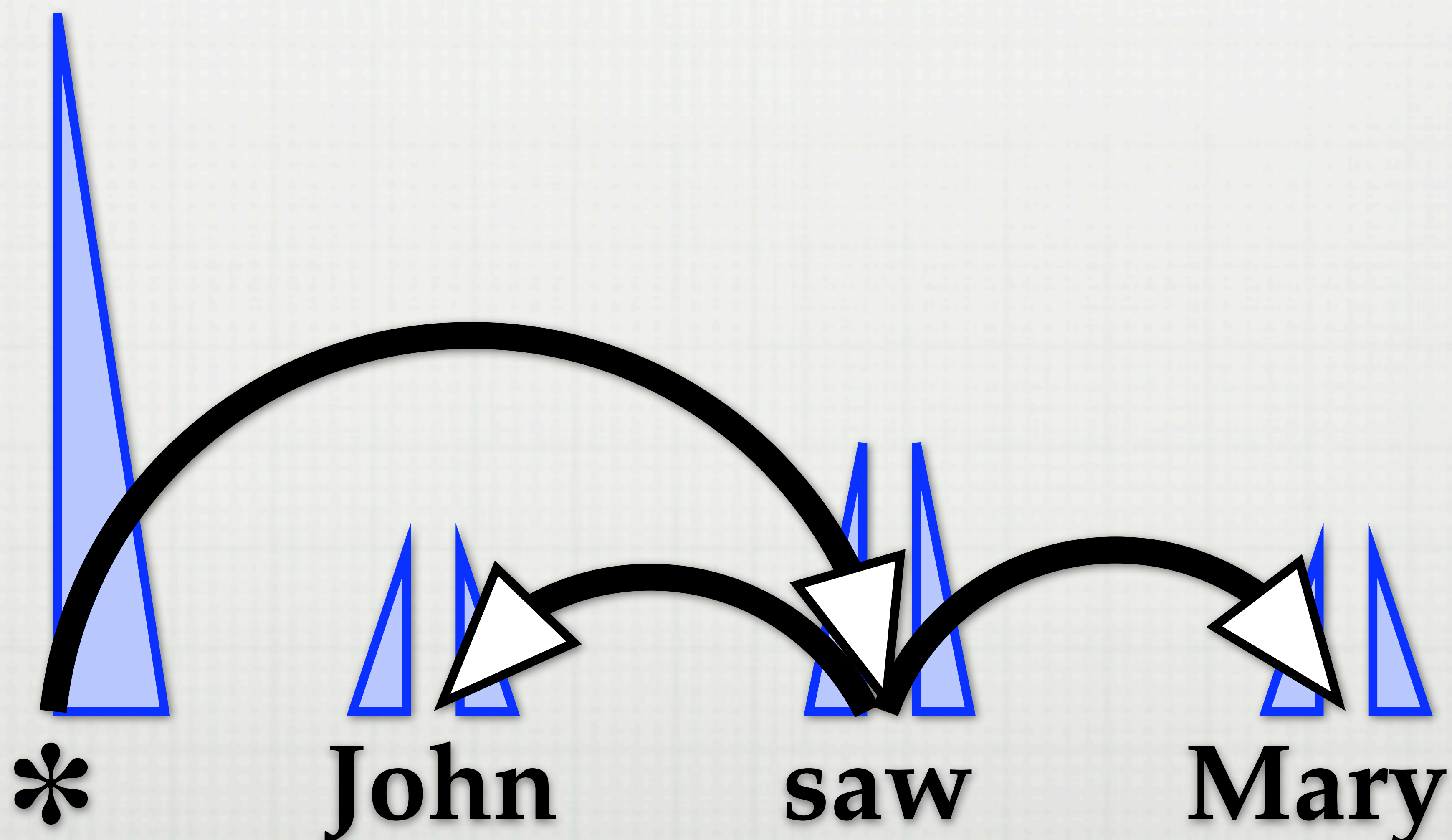
First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$



First-Order Parsing Example

- Eisner (2000) algorithm: $O(n^3)$

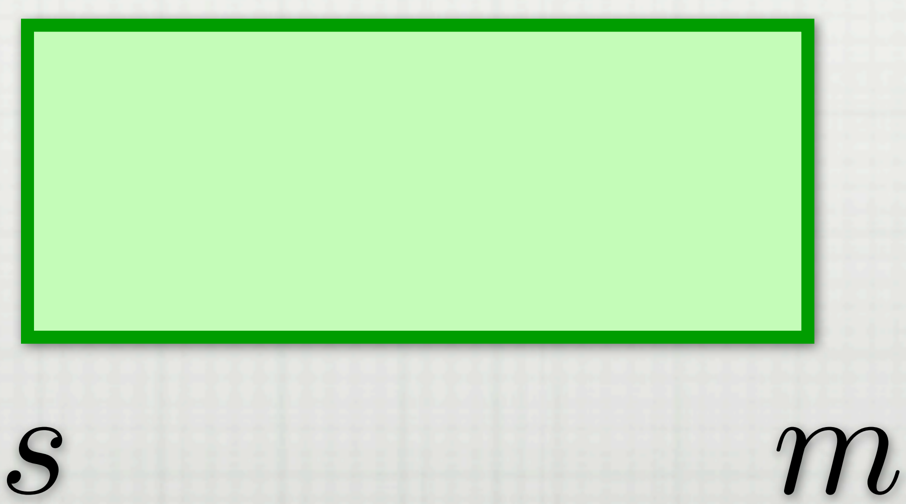


Second-Order Sibling Parser

- McDonald (2006) and Eisner (1996): $O(n^3)$
  - Introduce a third type of span:

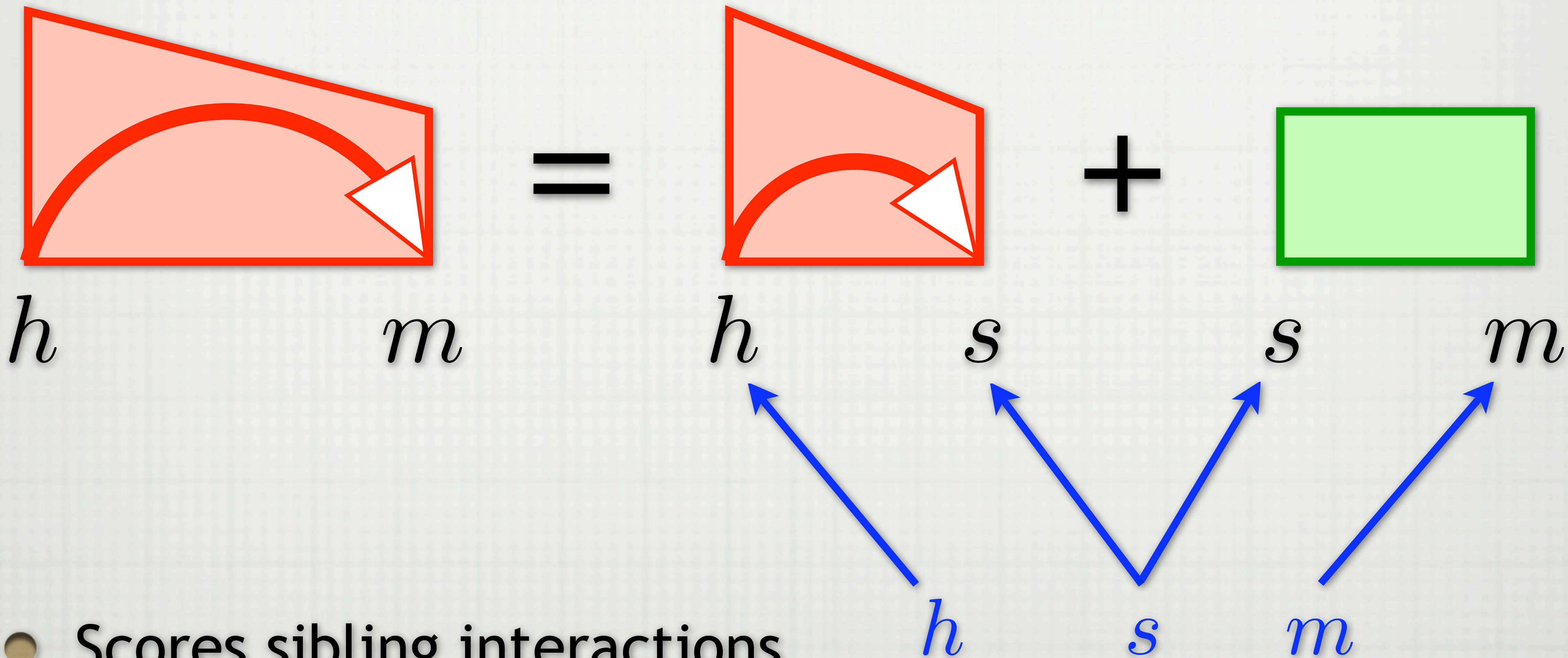
Sibling Span

A pair of adjacent modifiers



Second-Order Sibling Parser

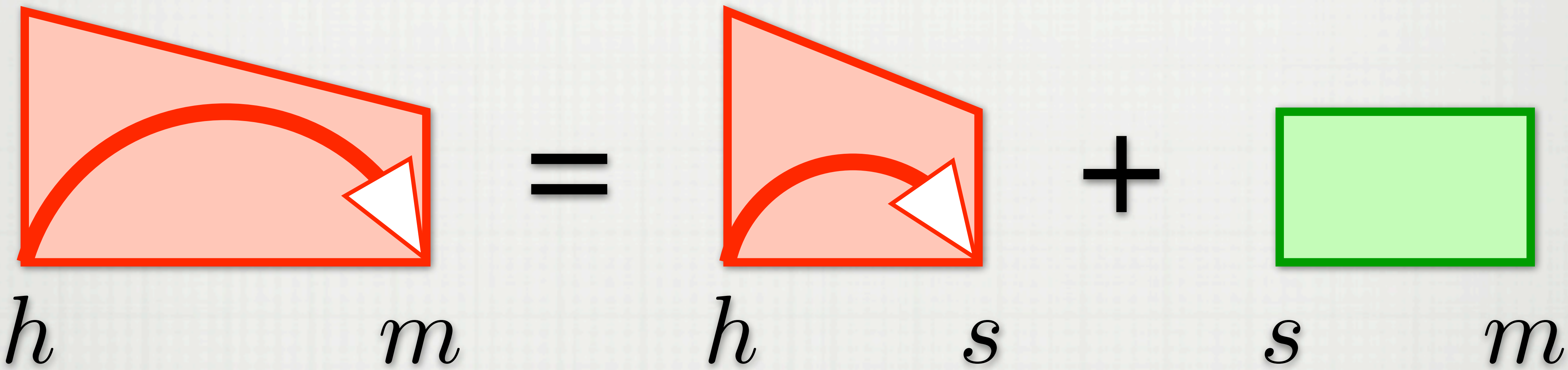
- McDonald (2006) and Eisner (1996): $O(n^3)$



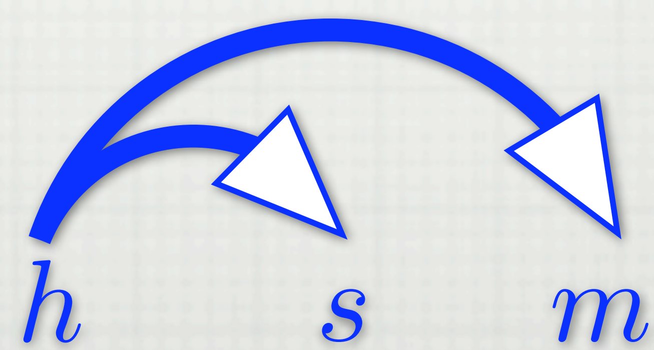
- Scores sibling interactions

Second-Order Sibling Parser

- McDonald (2006) and Eisner (1996): $O(n^3)$

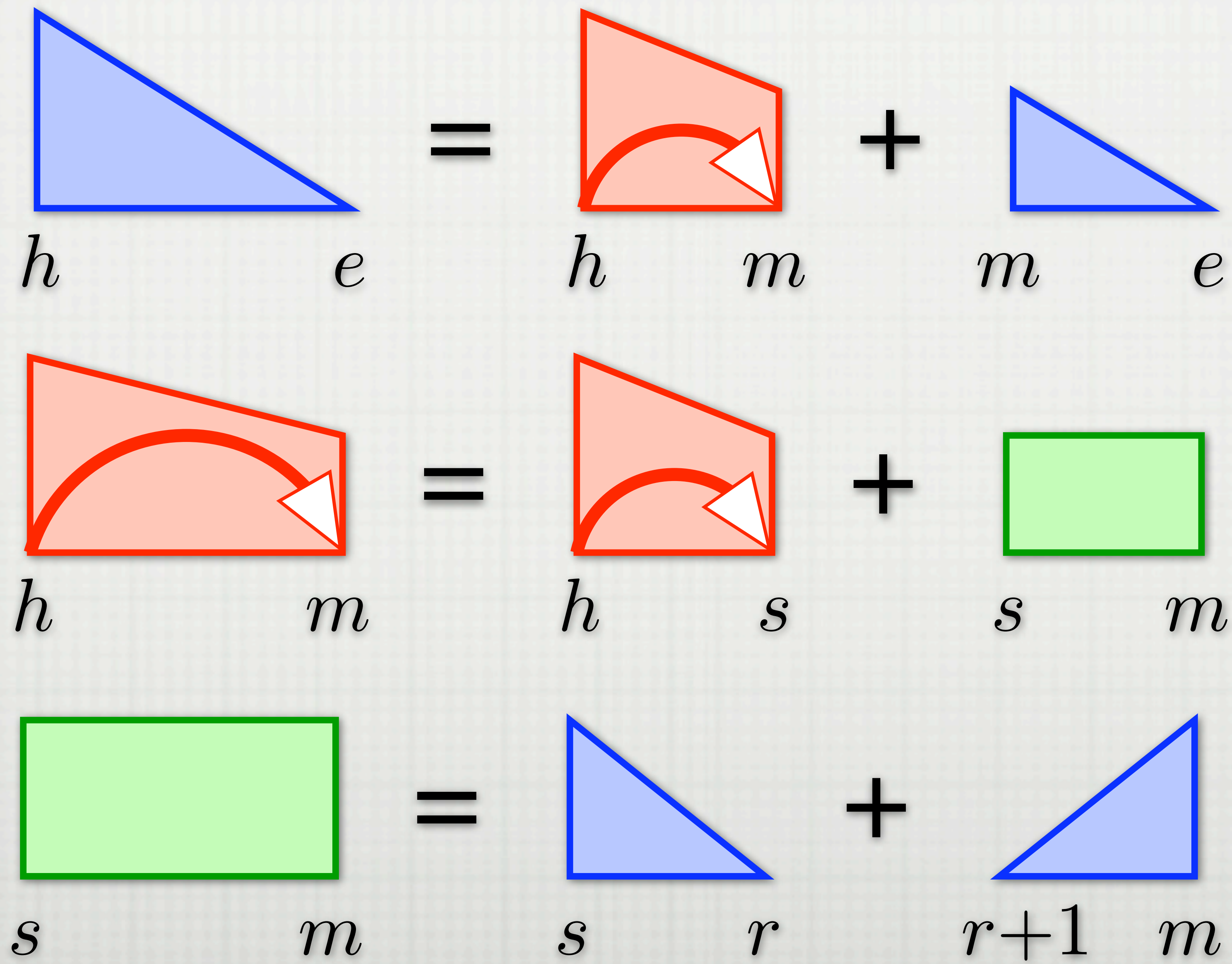


- Scores sibling interactions



Second-Order Sibling Parser

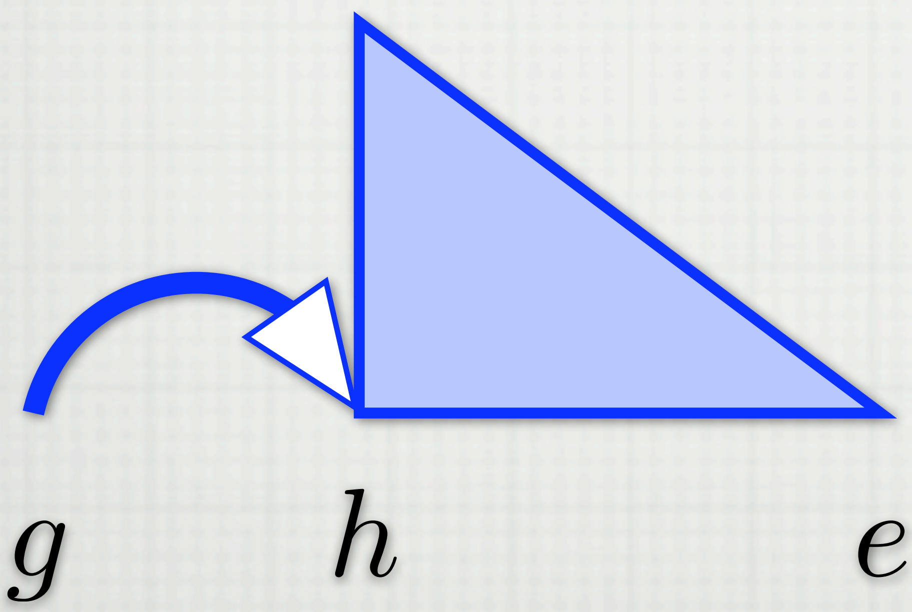
- McDonald (2006) and Eisner (1996): $O(n^3)$



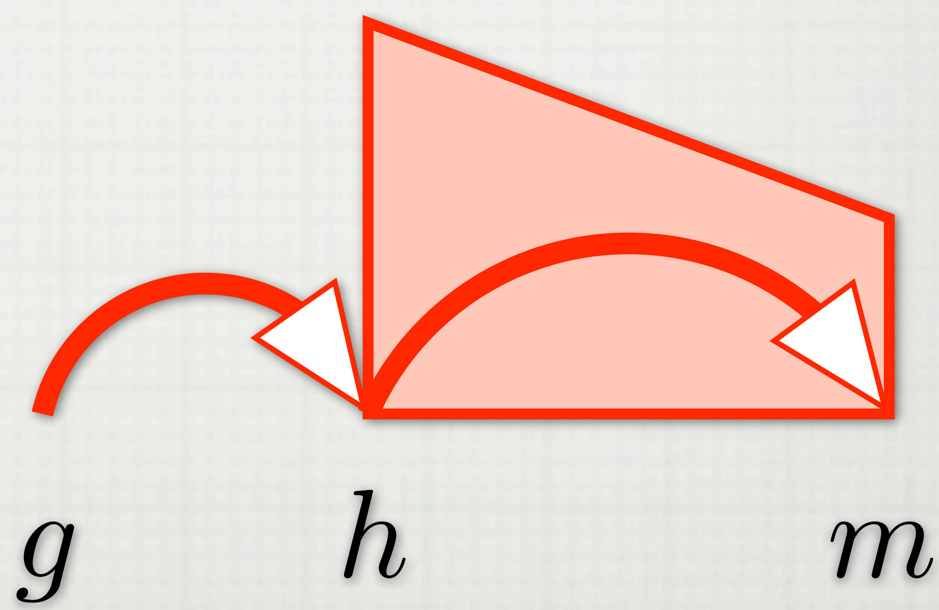
Model 0

- Model 0, all grandparents: $O(n^4)$

Complete G-Span
A "half-constituent"
with its grandparent



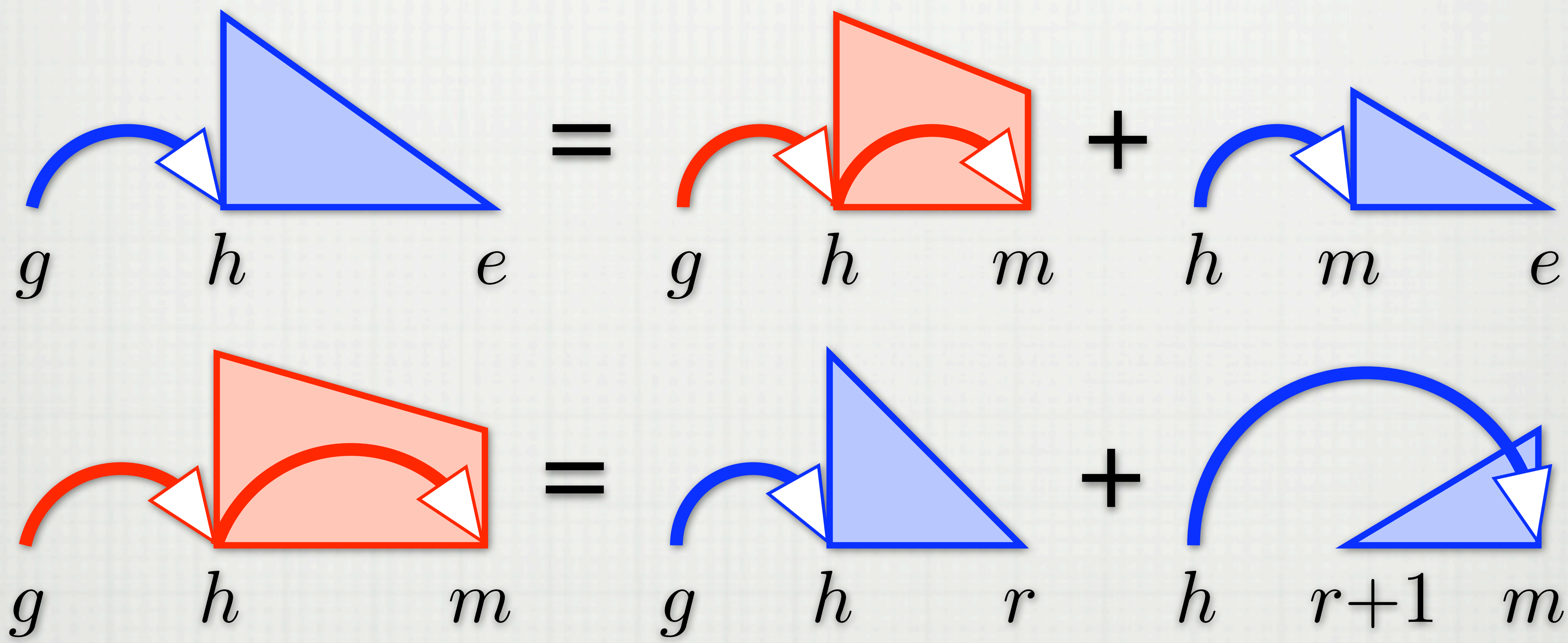
Incomplete G-Span
A dependency
with its grandparent



- Superficially similar to parent annotation in CFGs

Model 0: Derivations

- Model 0, all grandparents: $O(n^4)$



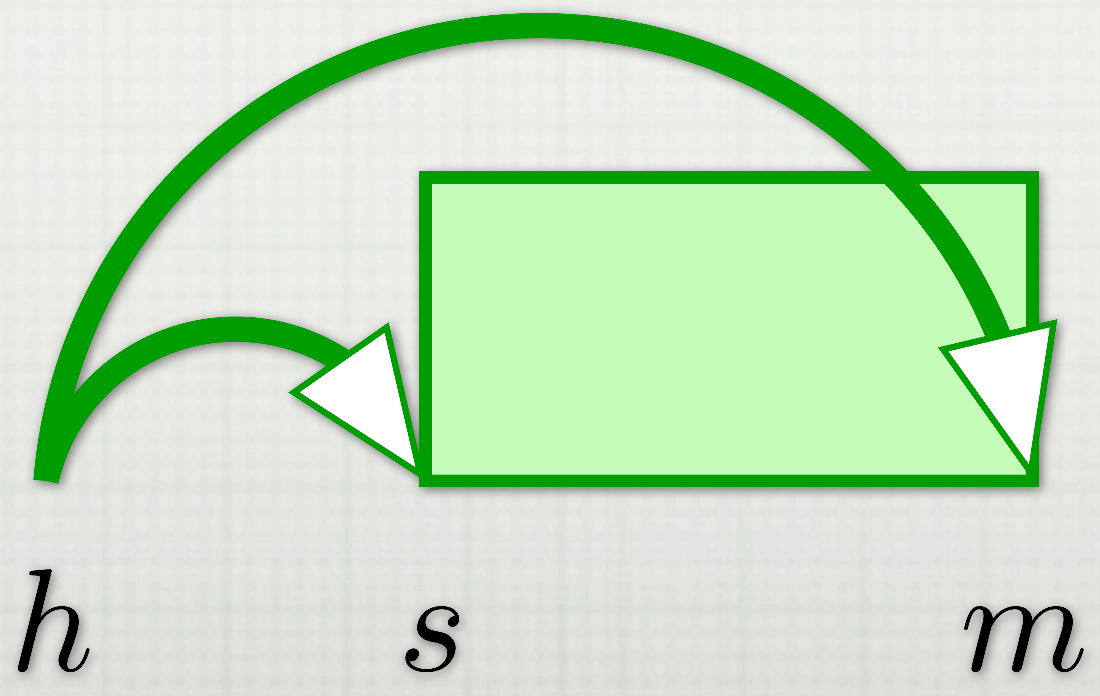
- Grandparent indices propagated to smaller g-spans
- 4 active indices, runtime $O(n^4)$

Model 1

- Model 1, grand-siblings: $O(n^4)$
  - Introduce a third type of span:

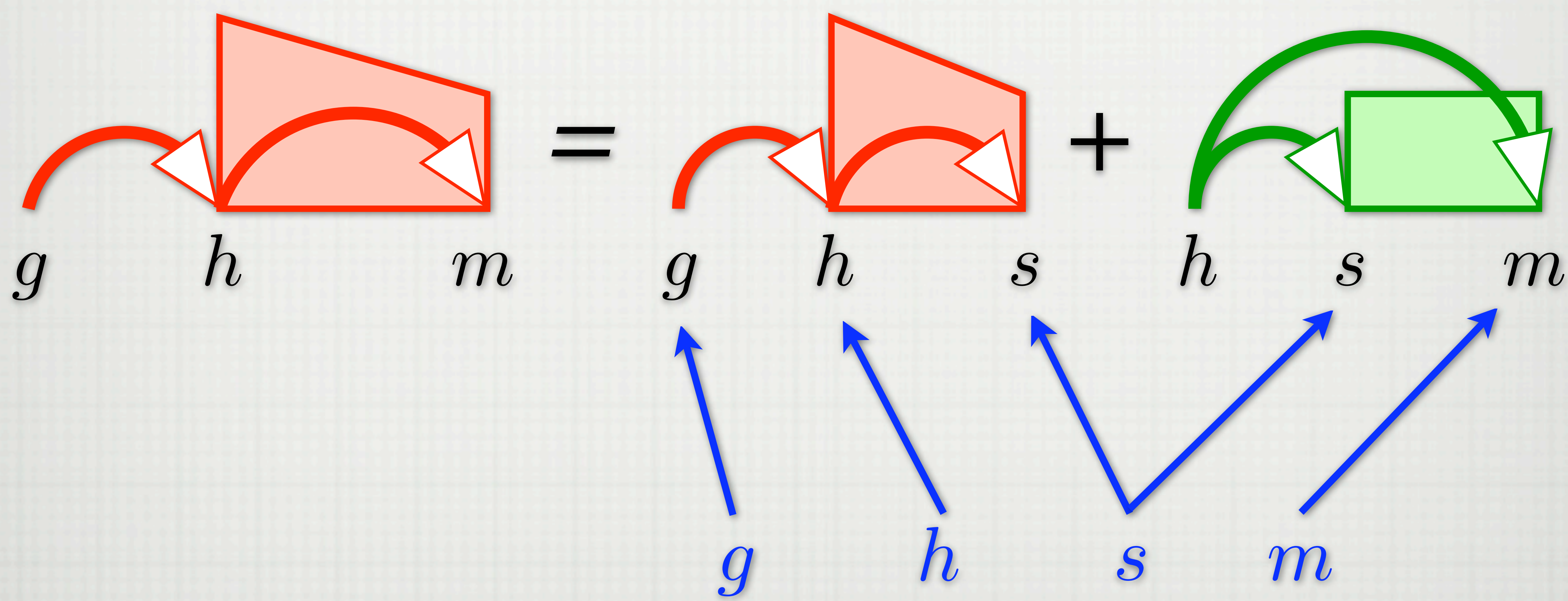
Sibling G-Span

A pair of adjacent modifiers with their shared head



Model 1: Grand-Sibling Scores

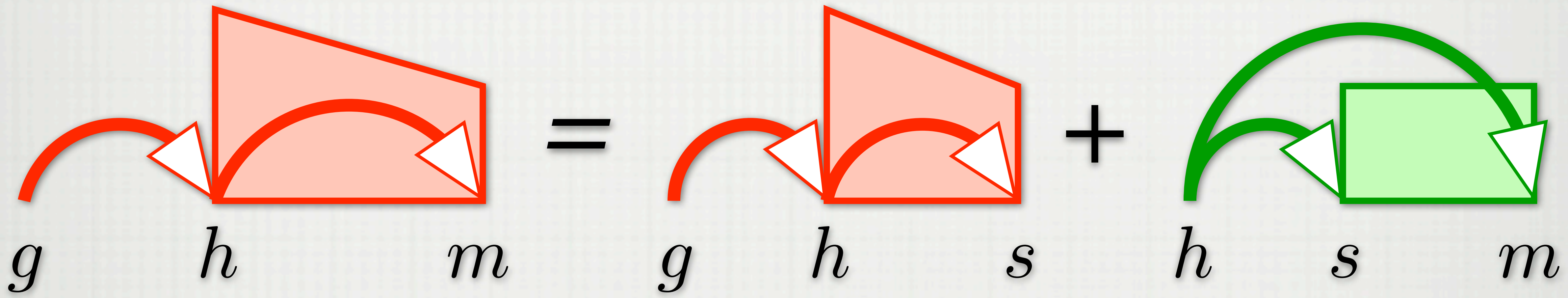
- Model 1, grand-siblings: $O(n^4)$

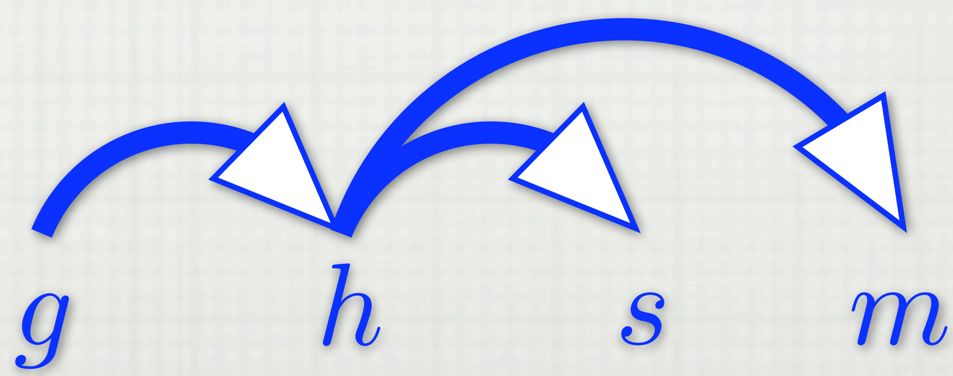


- Scores grand-sibling interactions

Model 1: Grand-Sibling Scores

- Model 1, grand-siblings: $O(n^4)$

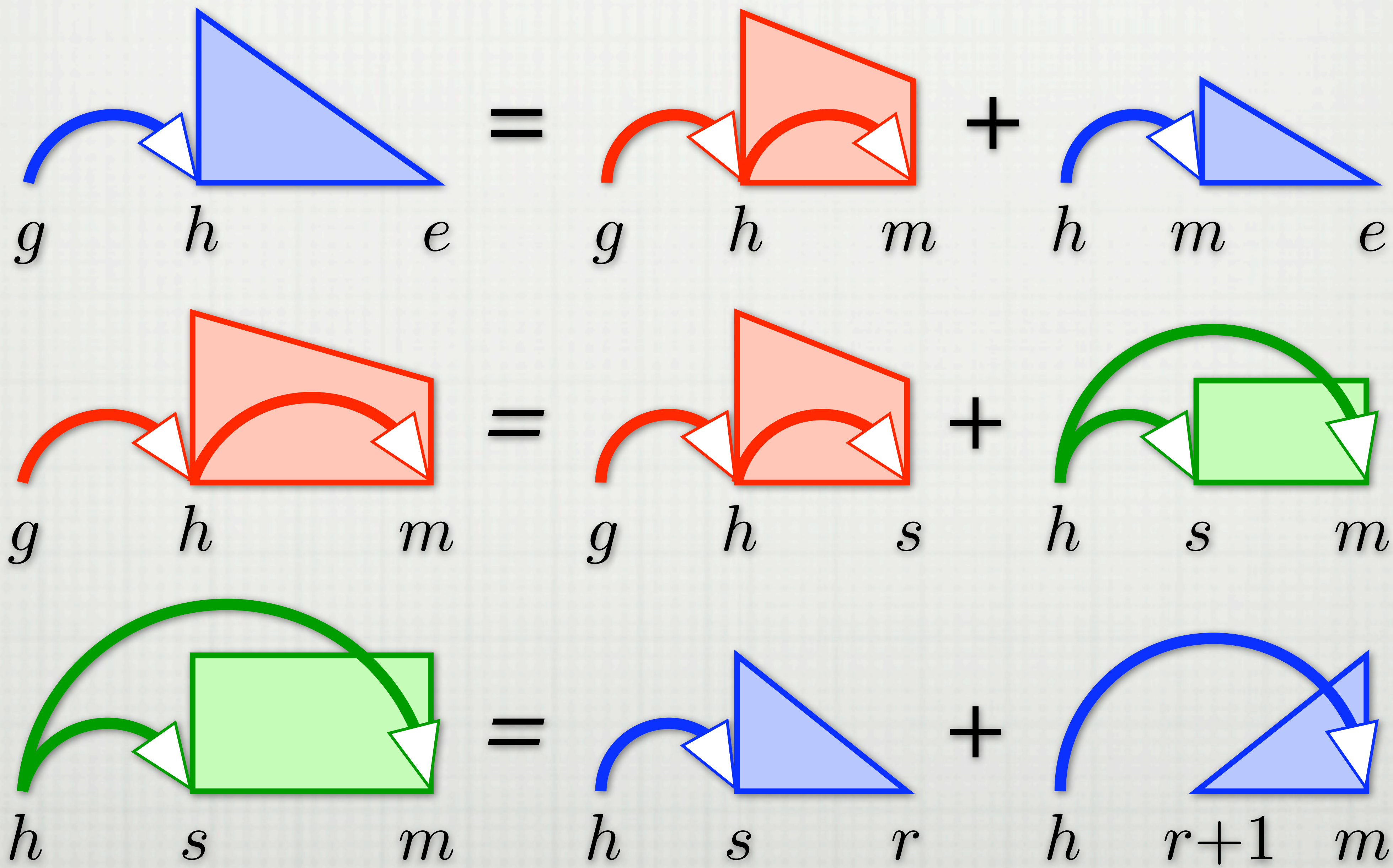




- Scores grand-sibling interactions

Model 1: Derivations

- Model 1, grand-siblings: $O(n^4)$

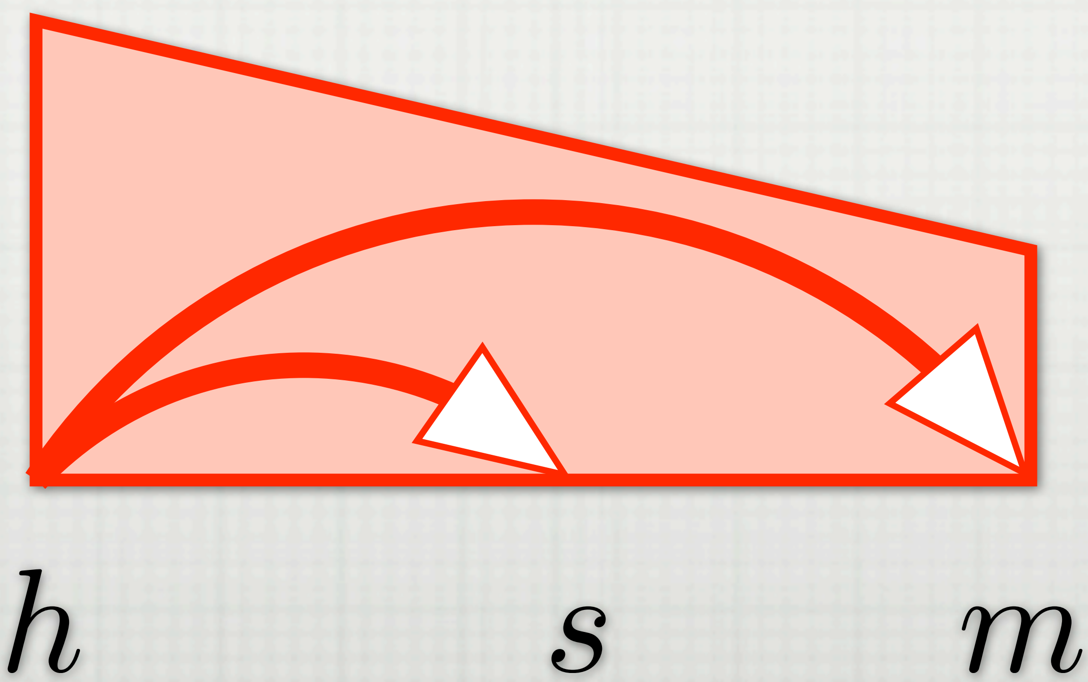


Model 2

- Model 2, grand-siblings and tri-siblings: $O(n^4)$
  - Introduce a fourth type of span:

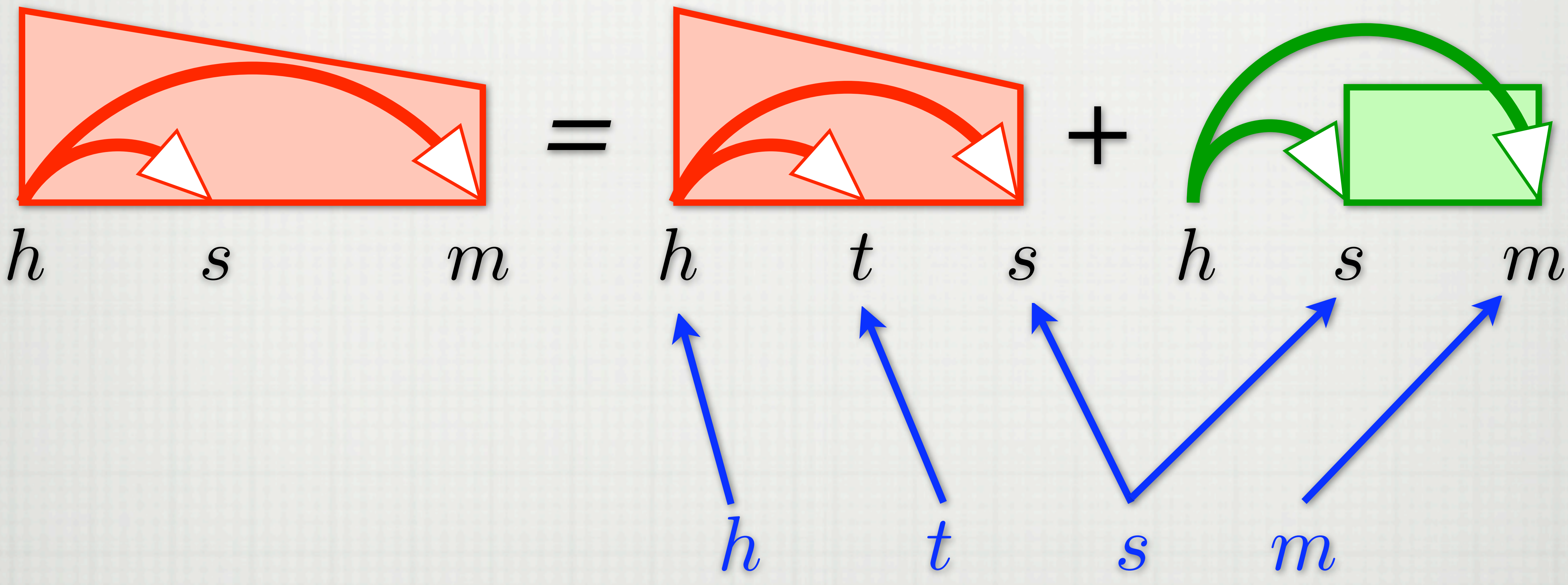
Incomplete S-Span

A dependency with its next-inner modifier



Model 2: Tri-Sibling Scores

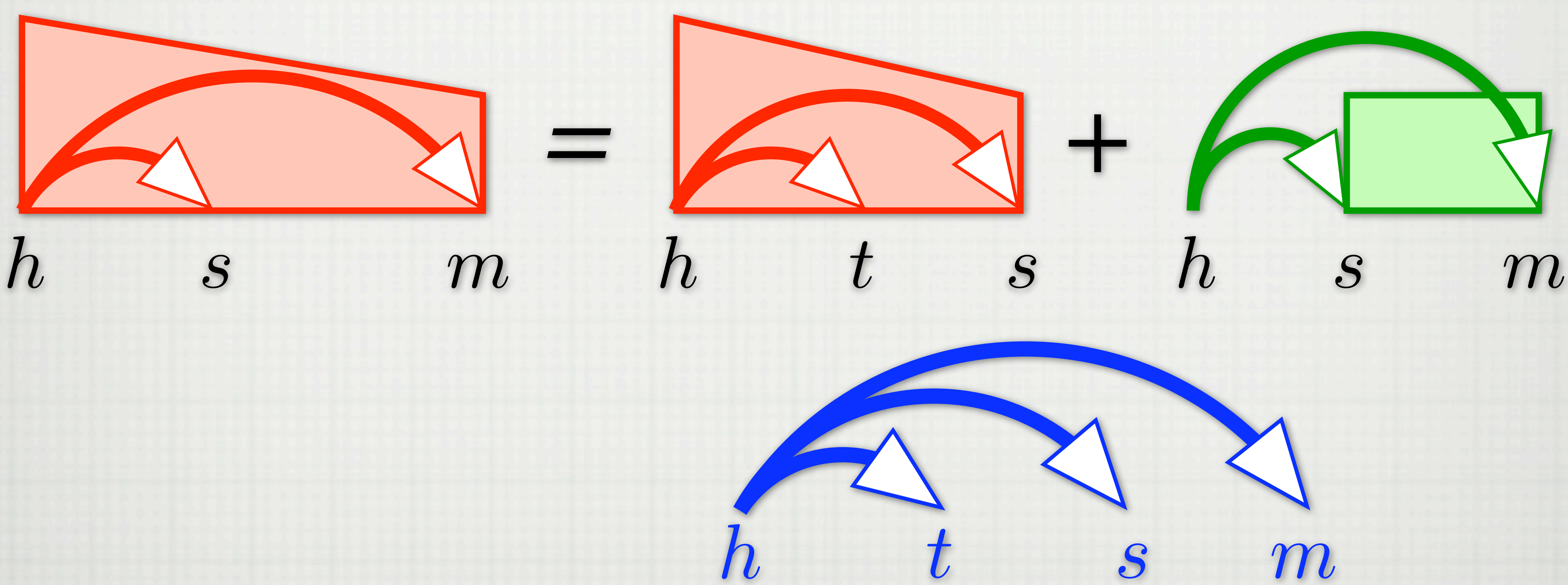
- Model 2, grand-siblings and tri-siblings: $O(n^4)$



- Scores tri-sibling interactions

Model 2: Tri-Sibling Scores

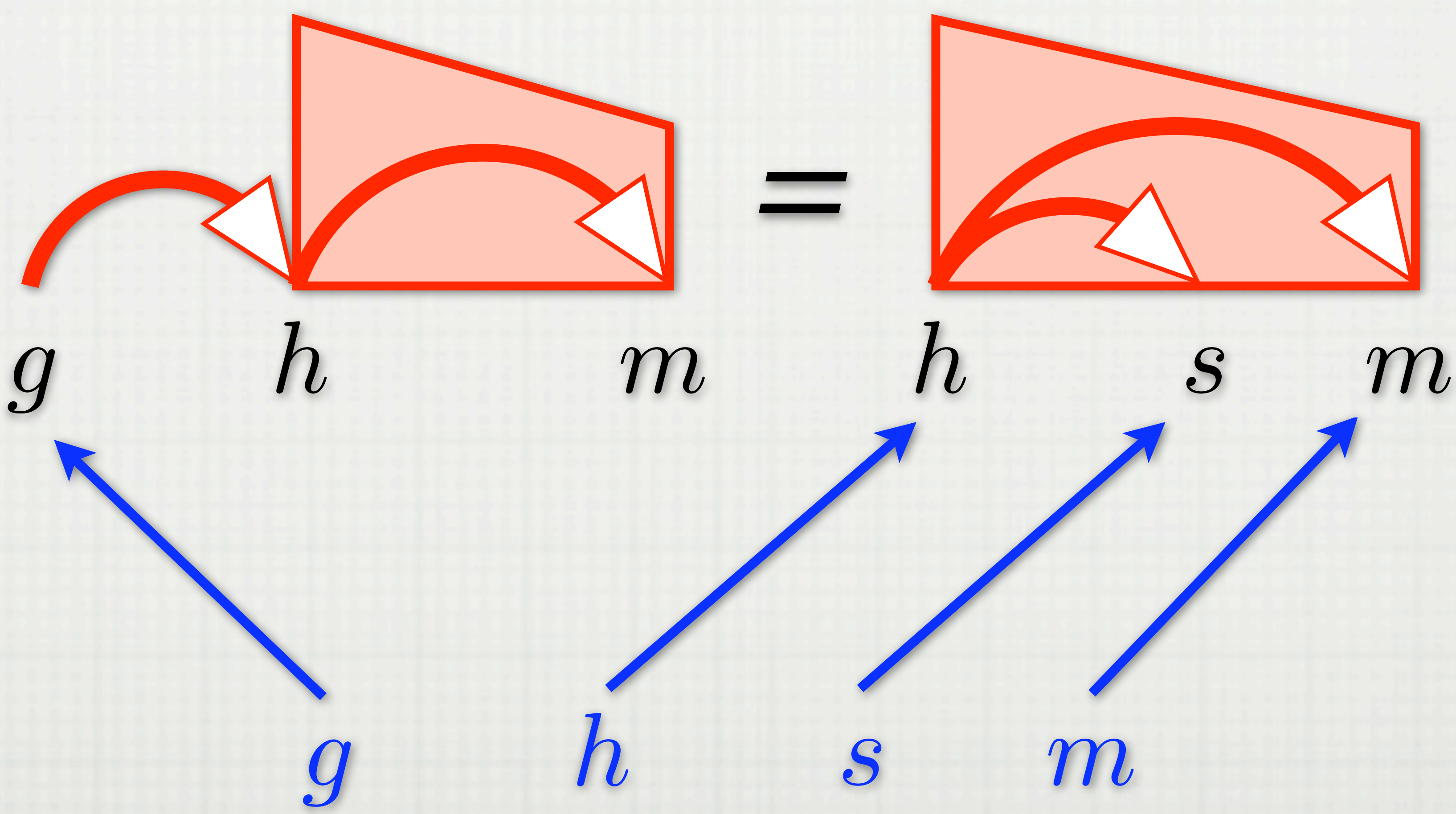
- Model 2, grand-siblings and tri-siblings: $O(n^4)$



- Scores tri-sibling interactions

Model 2: Grand-Sibling Scores

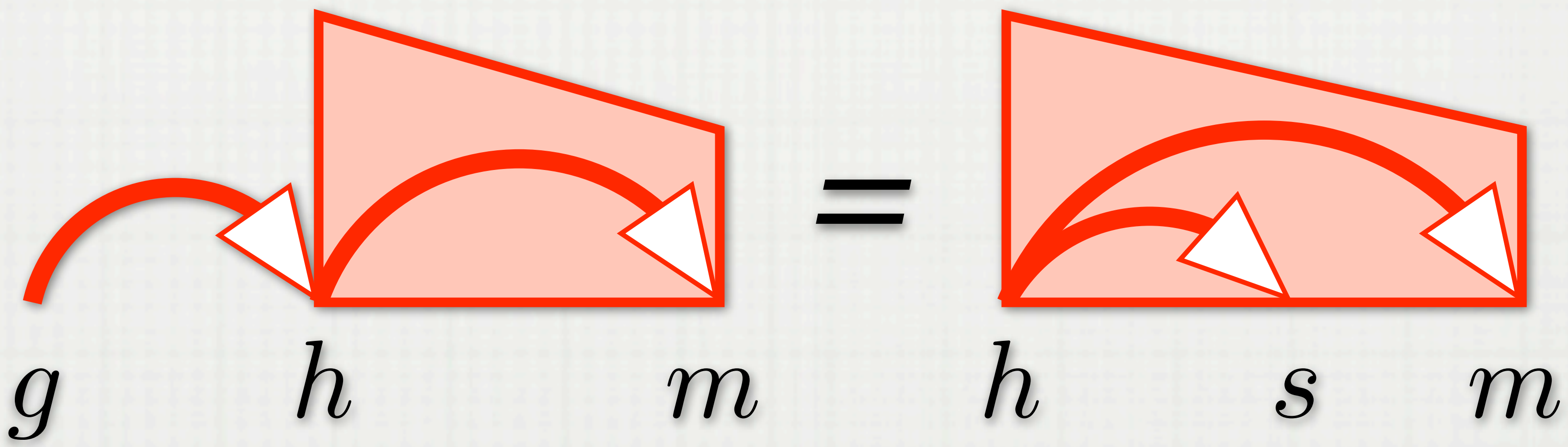
- Model 2, grand-siblings and tri-siblings: $O(n^4)$

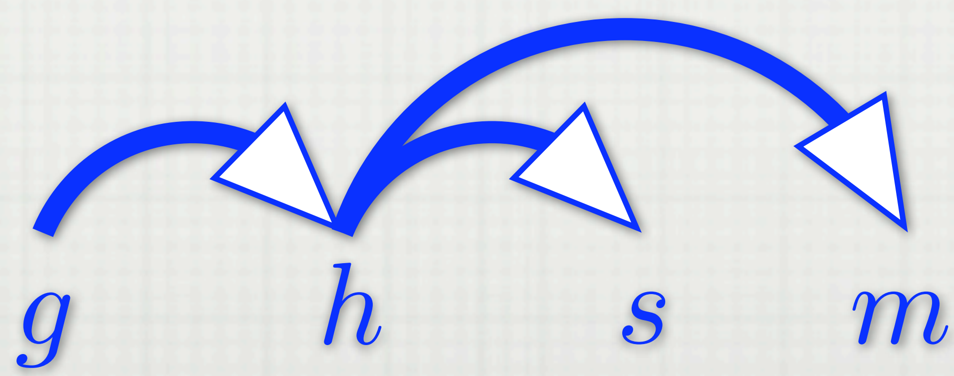


- Scores grand-sibling interactions

Model 2: Grand-Sibling Scores

- Model 2, grand-siblings and tri-siblings: $O(n^4)$

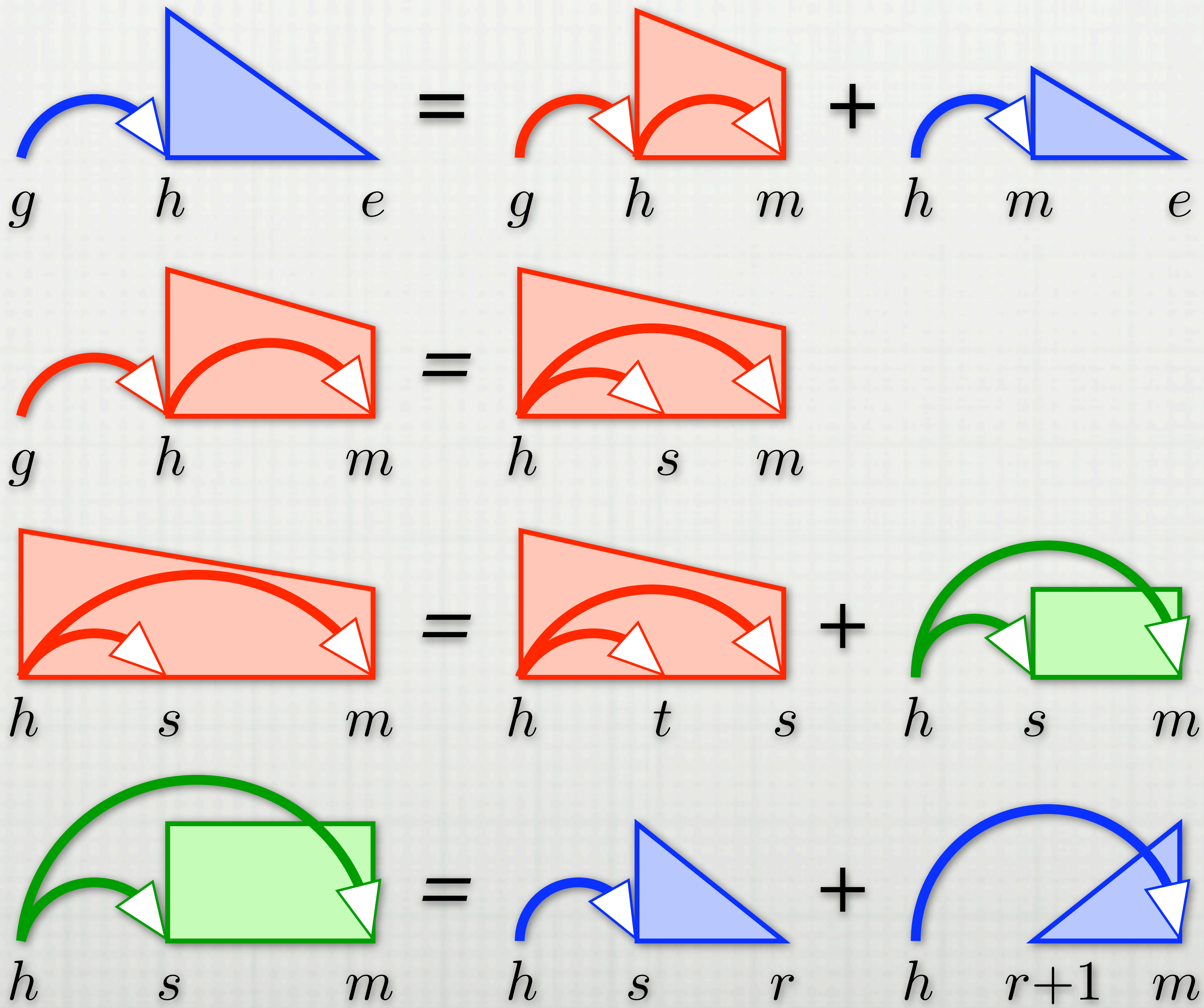




- Scores grand-sibling interactions

Model 2: Derivations

- Model 2, grand-siblings and tri-siblings: $O(n^4)$



Summary of Parsing Algorithms

- Model 0 parses an all-grandchildren factorization
- Model 1 parses an all-grand-siblings factorization
- Model 2 parses all-tri-siblings and some grand-siblings
- All parsers require $O(n^4)$ time and $O(n^3)$ space
  - Identical to Carreras (2007) second-order
- Models 1 and 2 are asymptotically fast:
  - Number of third-order parts is $\Omega(n^4)$

Parsing Experiments

- English Penn Treebank (Penn2Malt conversion)
- Czech Prague Dependency Treebank
- Averaged perceptron training
- Features based on words and POS tags
- Coarse-to-fine pruning (Carreras et al., 2008)

English and Czech Parsing

Parser	English	Czech
McDonald and Pereira (2006)	91.5	85.2
Koo, Carreras and Collins (2008), Normal	92.0	86.1
Model 1	93.0	87.4
Model 2	92.9	87.4
Koo, Carreras and Collins (2008), Semisup	93.2	87.1

- Unlabeled attachment score on the test sets
- Third-order is comparable to semi-supervised features

Final Remarks

- Third-order factorizations can be parsed in $O(n^4)$
- Third-parsers work well in practice
- Possible extensions:
  - Recovering word senses or dependency labels
  - Full head automata: e.g, TAG-style parsing (Carreras et al., 2008)
  - Increasing context to fourth-order or more