Lecture 7: Introduction to syntax-based MT

Andreas Maletti

Statistical Machine Translation

Stuttgart — December 16, 2011

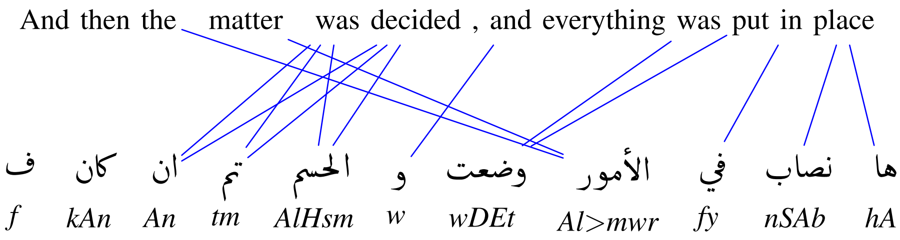
Lecture 7

Goals

- Overview
- Tree substitution grammars (tree automata)
- Synchronous grammars (tree transducers)

Contents

Word-based system (FST)



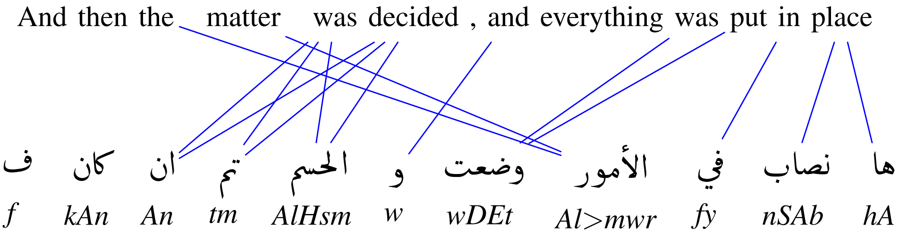
Derivation

Input:

[And] *then the matter was decided , and everything was put in place*

Output:

Word-based system (FST)



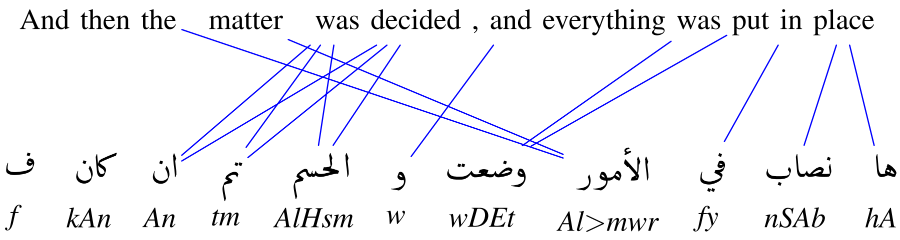
Derivation

Input:

then *the matter was decided , and everything was put in place*

Output:

Word-based system (FST)



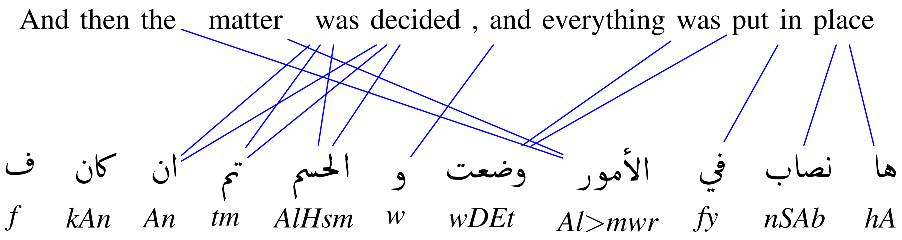
Derivation

Input:

the | *matter was decided , and everything was put in place*

Output:

Word-based system (FST)



Derivation

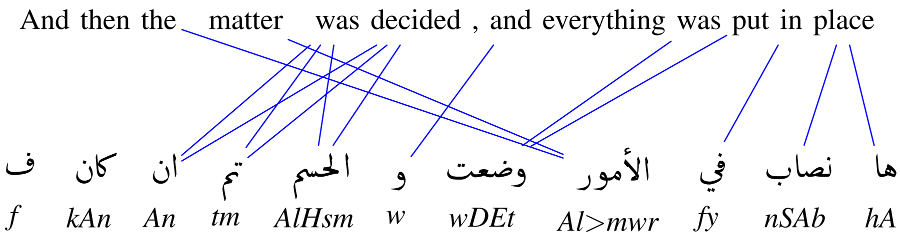
Input:

the *matter was decided , and everything was put in place*

Output:

f

Word-based system (FST)



Derivation

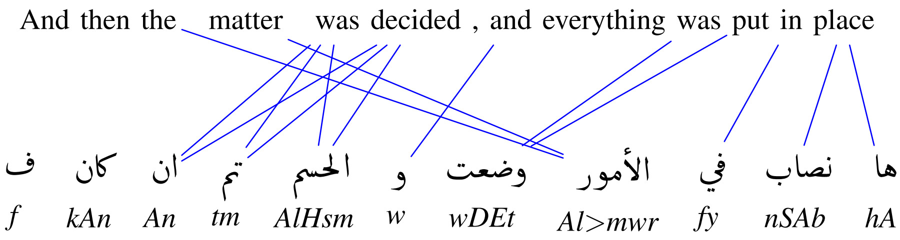
Input:

the *matter was decided , and everything was put in place*

Output:

f kAn

Word-based system (FST)



Derivation

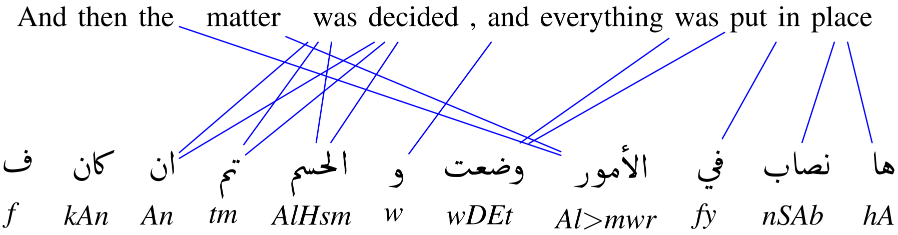
Input:

the *matter* *was decided , and everything was put in place*

Output:

f kAn

Word-based system (FST)



Derivation

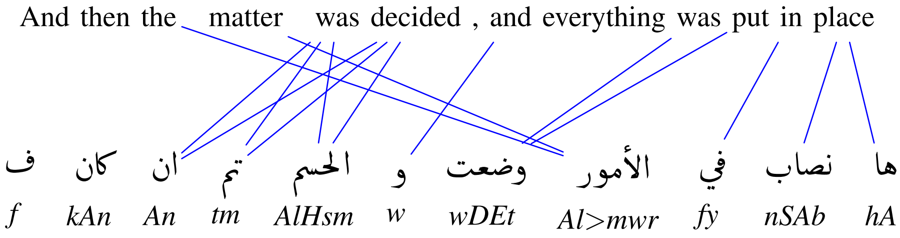
Input:

the matter [*was*] *decided , and everything was put in place*

Output:

f kAn

Word-based system (FST)



Derivation

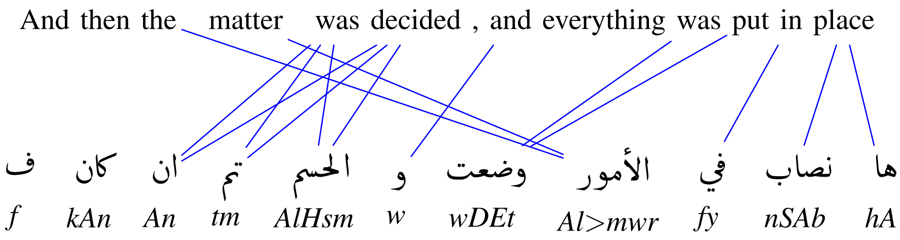
Input:

the matter was *decided* *, and everything was put in place*

Output:

f kAn

Word-based system (FST)



Derivation

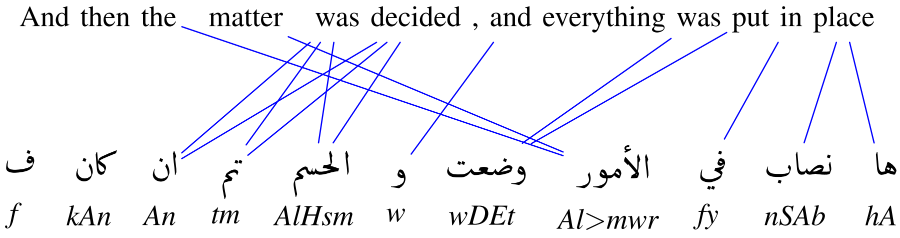
Input:

the matter | , | *and everything was put in place*

Output:

f kAn An tm AlHsm

Word-based system (FST)



Derivation

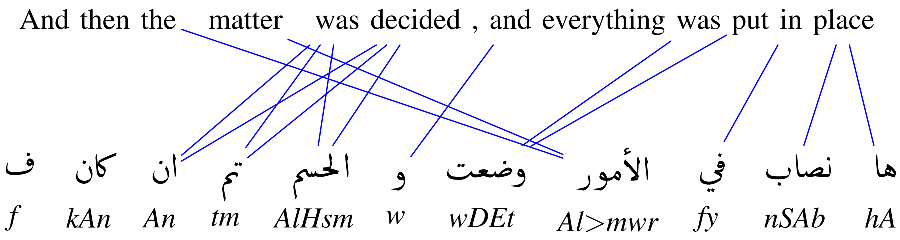
Input:

the matter [*and*] *everything was put in place*

Output:

f kAn An tm AlHsm

Word-based system (FST)



Derivation

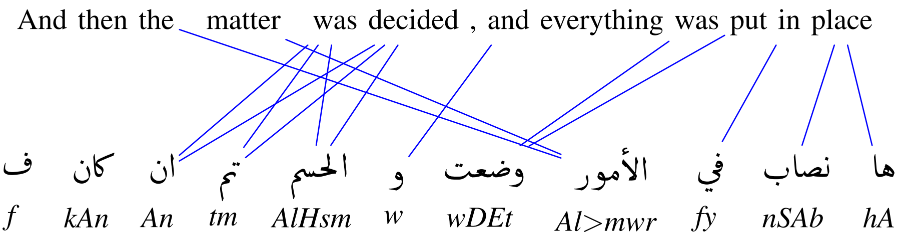
Input:

the matter *everything* *was put in place*

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

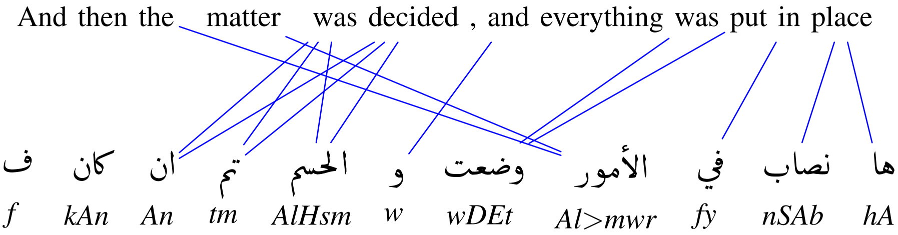
Input:

the matter *was* *put in place*

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

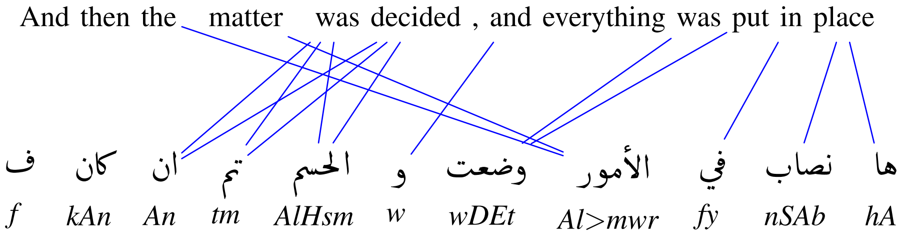
Input:

the matter was [put] in place

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

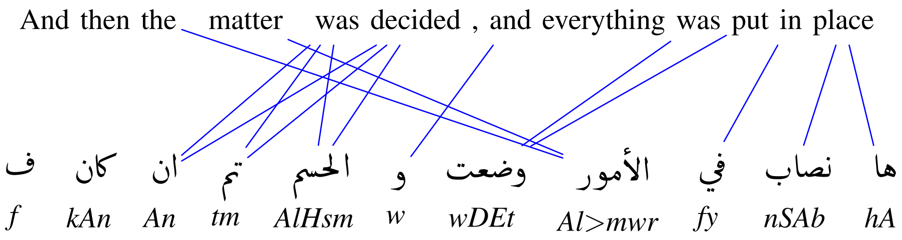
Input:

the matter *in* *place*

Output:

f kAn An tm AlHsm w wDEt

Word-based system (FST)



Derivation

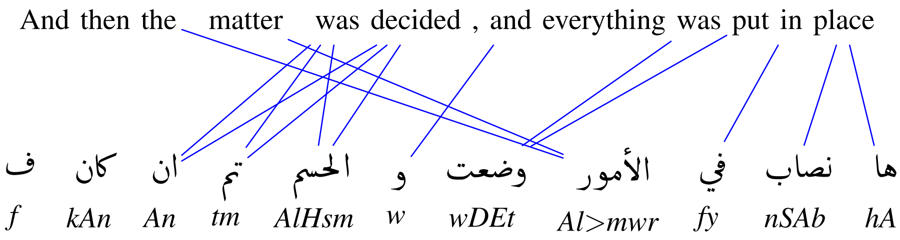
Input:

[*in*] *place*

Output:

f kAn An tm AlHsm w wDEt Al>mwr

Word-based system (FST)



Derivation

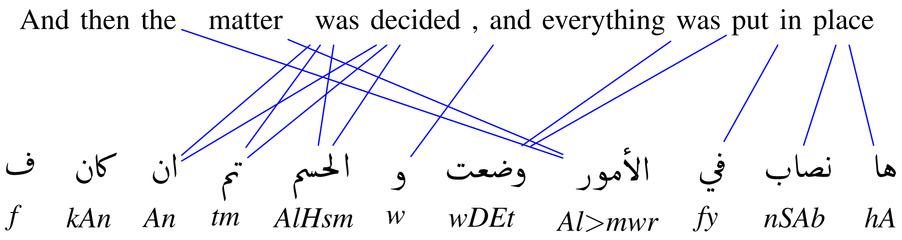
Input:

place

Output:

f kAn An tm AlHsm w wDEt Al>mwr fy

Word-based system (FST)



Derivation

Input:

Output:

f kAn An tm AlHsm w wDEt Al>mwr fy nSAb hA

Phrase-based machine translation

Schema

Input ⟶ Machine translation system ⟶ Language model ⟶ Output

Phrase-based systems

Input ⟶ Segmenter ⟶ Machine translation system ⟶ Language model ⟶ Output

Phrase-based machine translation

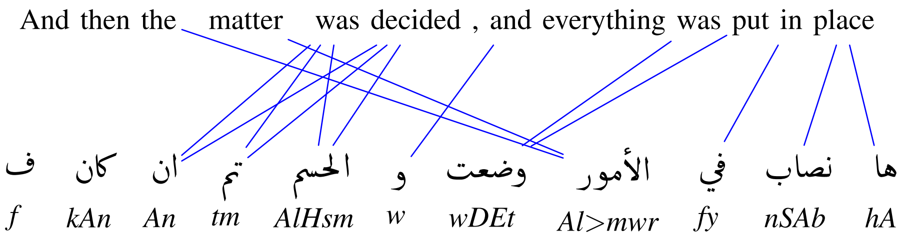
Schema

Input ⟶ Machine translation system ⟶ Language model ⟶ Output

Phrase-based systems

Input ⟶ Segmenter ⟶ Machine translation system ⟶ Language model ⟶ Output

Phrase-based system (FST+Perm)



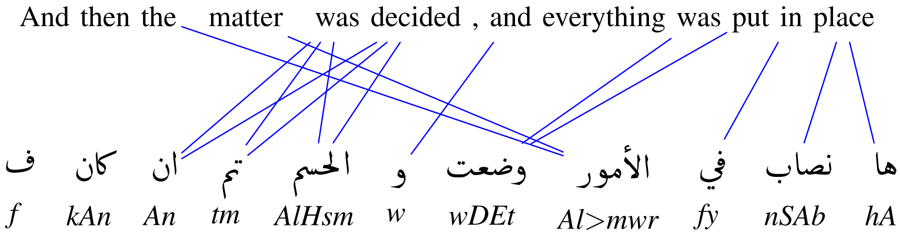
Derivation

Input:

And then the matter was decided , and everything was put in place

Output:

Phrase-based system (FST+Perm)



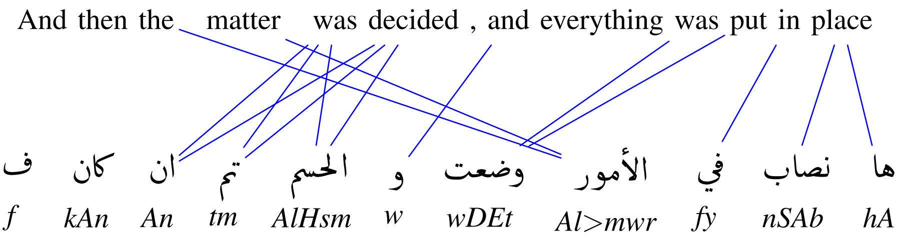
Derivation

Input:

And then 1 *the matter* 5 *was decided* 2 *, and everything* 3 *was put* 4 *in place* 6

Output:

Phrase-based system (FST+Perm)



Derivation

Input:

And then 1 *the matter* 5 *was decided* 2 *, and everything* 3 *was put* 4 *in place* 6

Output:

f kAn 1 *An tm AlHsm* 2 *w* 3 *wDEt* 4 *Almwr* 5 *fy nSAb hA* 6

Machine translation (cont'd)

Phrase-based systems

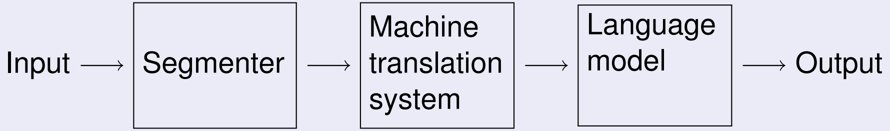
Input ⟶ Segmenter ⟶ Machine translation system ⟶ Language model ⟶ Output

Syntax-based systems

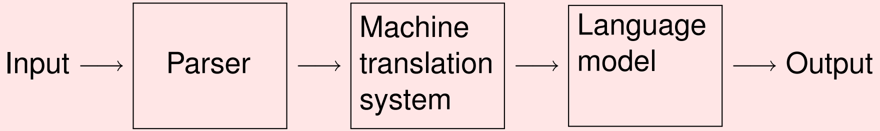
Input ⟶ Parser ⟶ Machine translation system ⟶ Language model ⟶ Output

Machine translation (cont'd)

Phrase-based systems

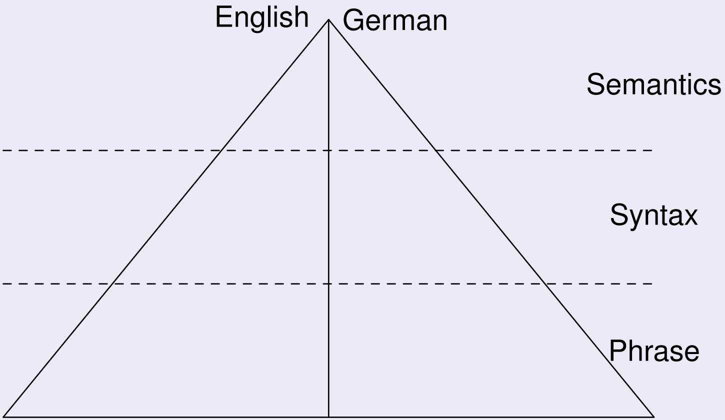


Syntax-based systems



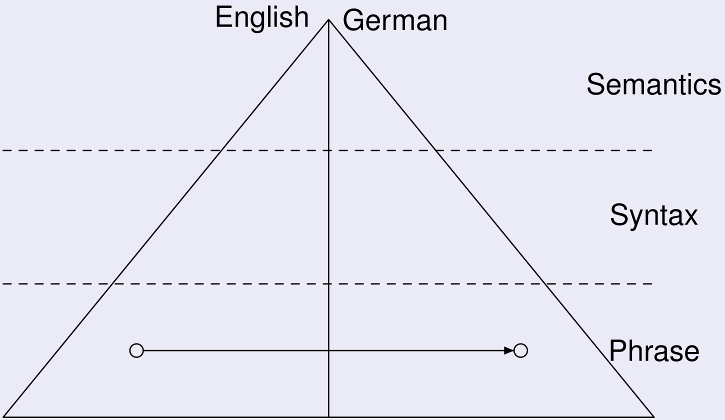
Syntax-based Approach

Overview



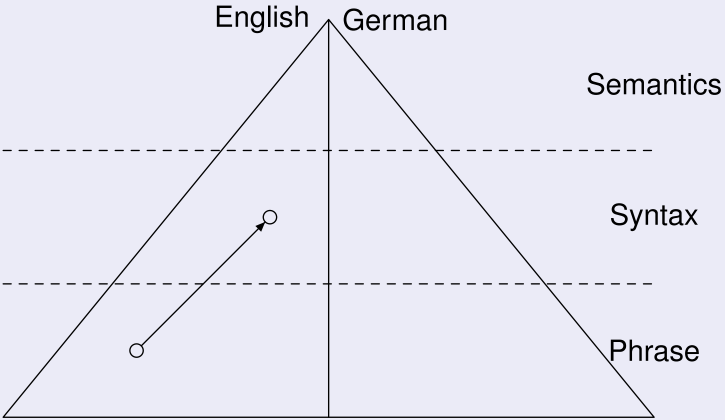
Syntax-based Approach

Overview



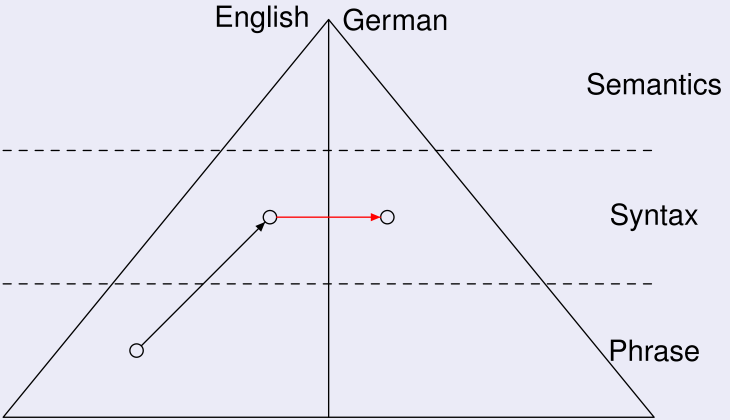
Syntax-based Approach

Overview



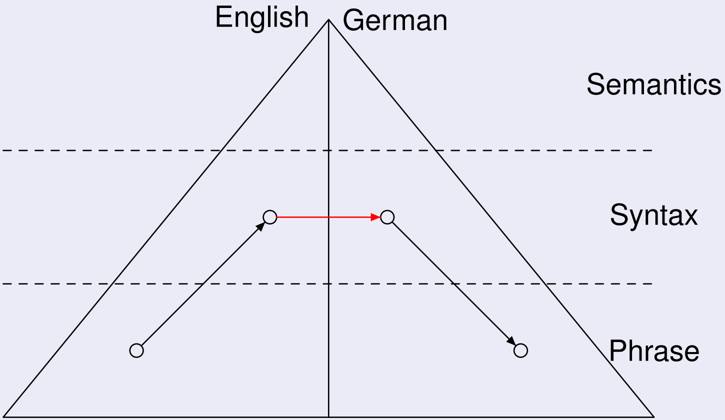
Syntax-based Approach

Overview

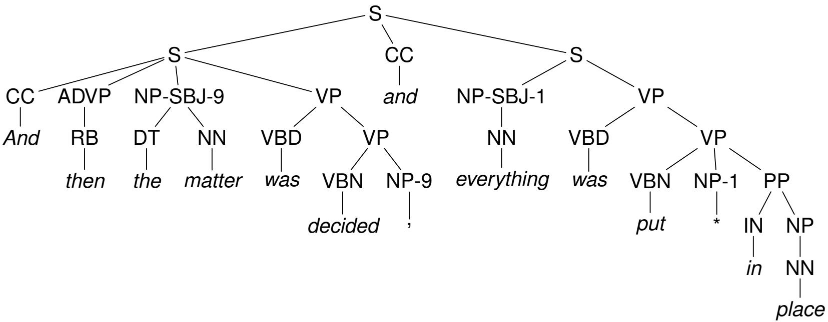


Syntax-based Approach

Overview



Parser

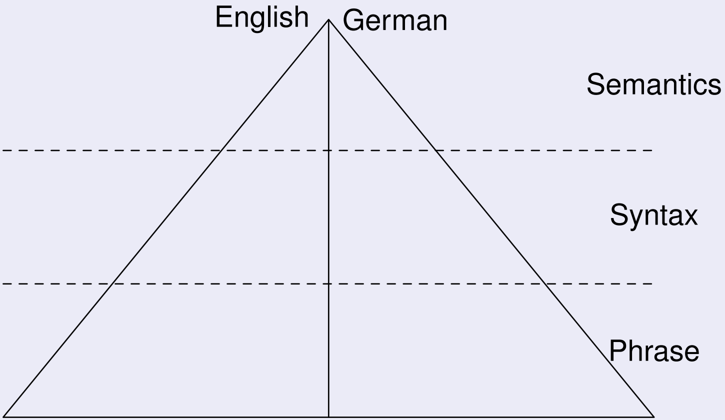


And then the matter was decided , and everything was put in place

(thanks to KEVIN KNIGHT for the data)

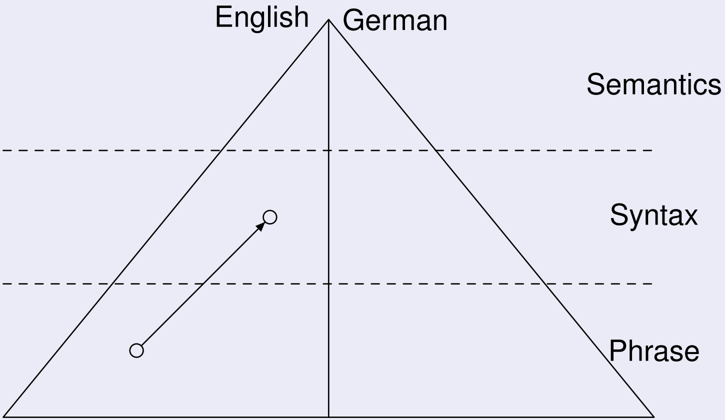
Semantics-based Approach

Overview



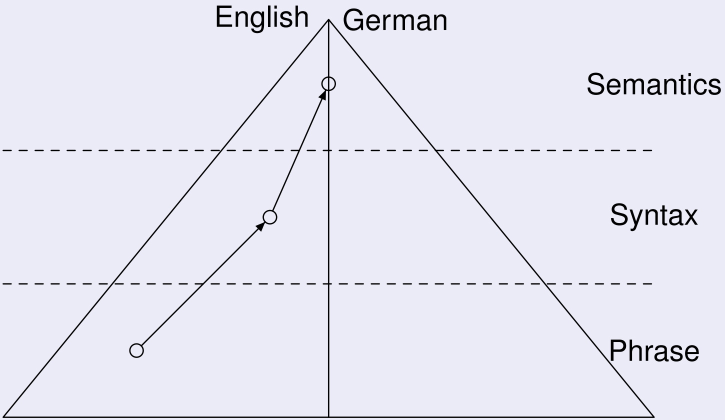
Semantics-based Approach

Overview



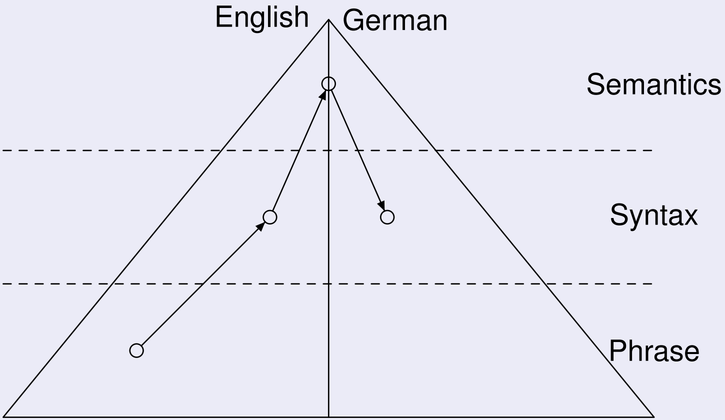
Semantics-based Approach

Overview



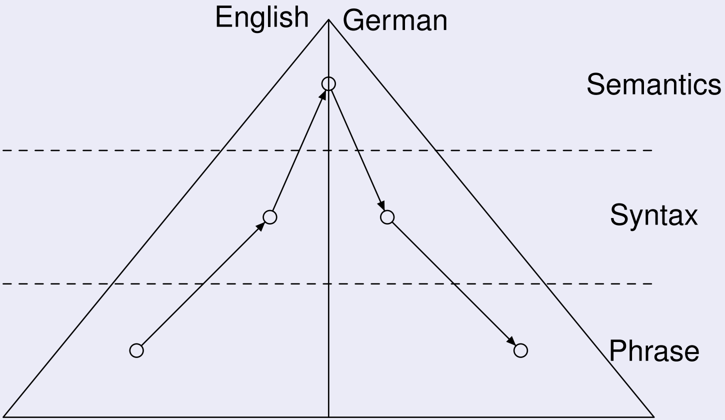
Semantics-based Approach

Overview



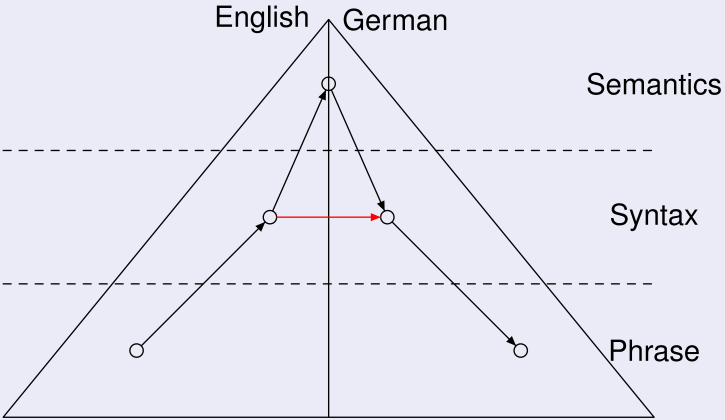
Semantics-based Approach

Overview



Semantics-based Approach

Overview



Contents

Parsing and CFG

Example (Context-free grammar)

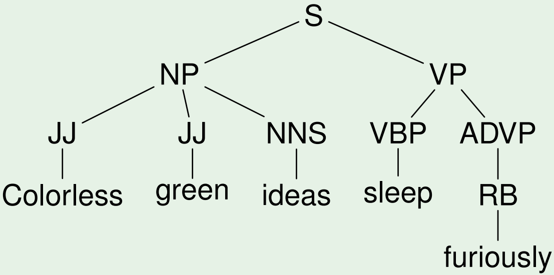
$$
\begin{aligned}
S &\to NP\ VP & NP &\to JJ\ JJ\ NNS \\
VP &\to VBP\ ADVP & ADVP &\to RB \\
JJ &\to \textit{Colorless} & JJ &\to \textit{green} \\
NNS &\to \textit{ideas} & VBP &\to \textit{sleep} \\
RB &\to \textit{furiously} & &
\end{aligned}
$$

Derivation

S $\to^*$ Colorless green ideas sleep furiously

Parse tree

Example

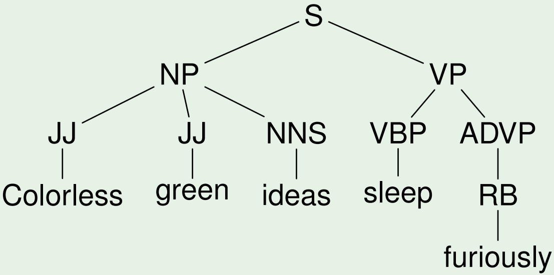


Remark

We are interested in the parse tree, not just whether $S \to^* w$!

Parse tree

Example

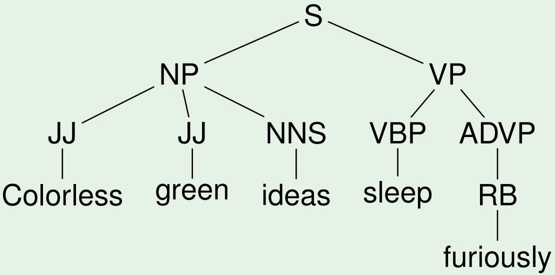


Remark

We are interested in the parse tree, not just whether S $\rightarrow^* w$!

Parse tree

Example



Remark

We are interested in the parse tree, not just whether S $\to^* w$!

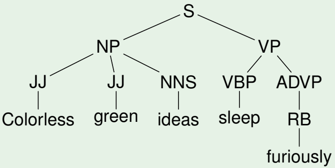
But there can be exponentially many parse trees for a sentence.

Packed tree language

Remark

A tree language is often called forest in NLP.

Example

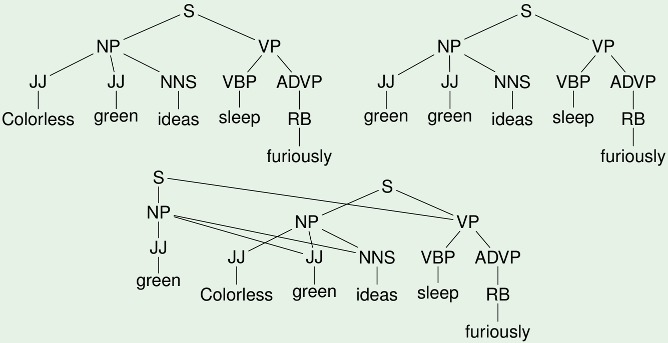


Packed tree language

Remark

A tree language is often called forest in NLP.

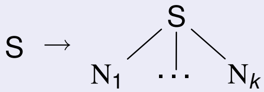
Example



Local tree language

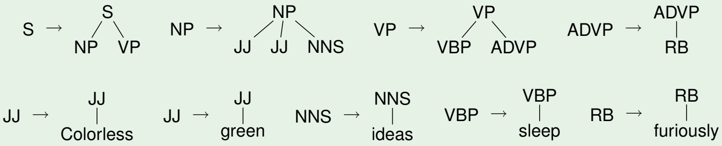
Definition

A local tree grammar is a grammar with rules of the form

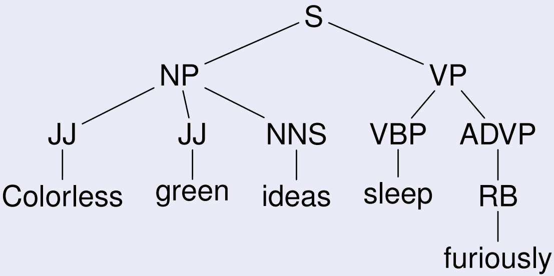


Local tree language

Example



Derivation



Local tree language

Definition

The tree languages generated by local tree grammars are the local tree languages.

Theorem

The set of derivations of a context-free grammar forms a local tree language.

Local tree language

Definition

The tree languages generated by local tree grammars are the local tree languages.

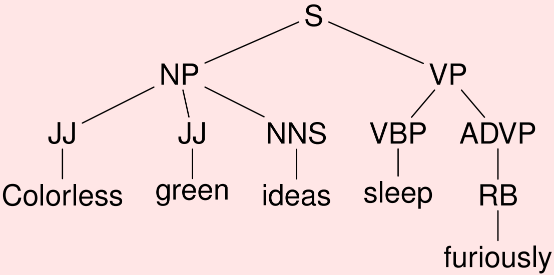
Theorem

The set of derivations of a context-free grammar forms a local tree language.

Local tree language

Question

Is the tree language consisting of only

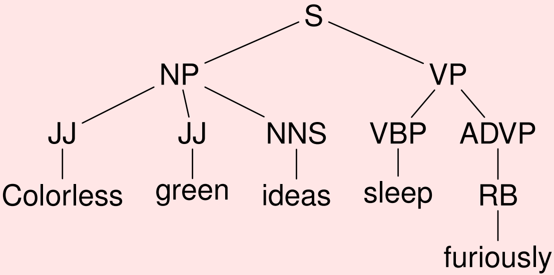


a local tree language?

Local tree language

Question

Is the tree language consisting of only



a local tree language?

Answer

NO!

Local tree language

Notes

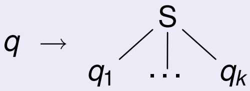
Local tree languages have undesirable properties:

- not closed under union
- cannot represent all finite languages
- ...

Regular tree language

Definition

A regular tree grammar is a grammar with rules of the form



The such generated languages are the regular tree languages.

Remark

Regular tree grammars are local tree grammars with hidden states.

Regular tree language

Definition

A regular tree grammar is a grammar with rules of the form

$$q \;\to\; \begin{array}{ccc} & S & \\ q_1 & \cdots & q_k \end{array}$$

The such generated languages are the regular tree languages.

Remark

Regular tree grammars are local tree grammars with hidden states.

Regular tree language

Example

$q_{\mathrm{S}} \to \mathrm{S}(q_{\mathrm{NP}}, q_{\mathrm{VP}})$ $\quad q_{\mathrm{NP}} \to \mathrm{NP}(p_{\mathrm{JJ}}, q_{\mathrm{JJ}}, q_{\mathrm{NNS}})$ $\quad q_{\mathrm{VP}} \to \mathrm{VP}(q_{\mathrm{VBP}}, q_{\mathrm{ADVP}})$ $\quad p_{\mathrm{JJ}} \to \mathrm{JJ}(q_c)$

$q_{\mathrm{JJ}} \to \mathrm{JJ}(q_g)$ $\quad q_{\mathrm{NNS}} \to \mathrm{NNS}(q_i)$ $\quad q_{\mathrm{VBD}} \to \mathrm{VBD}(q_s)$ $\quad q_{\mathrm{ADVP}} \to \mathrm{ADVP}(q_{\mathrm{RB}})$ $\quad q_{\mathrm{RB}} \to \mathrm{RB}(q_f)$

Derivation

$$q_{\mathrm{S}}$$

Regular tree language

Example

$q_S \rightarrow \mathrm{S}(q_{NP}, q_{VP})$ $\quad q_{NP} \rightarrow \mathrm{NP}(p_{JJ}, q_{JJ}, q_{NNS})$ $\quad q_{VP} \rightarrow \mathrm{VP}(q_{VBP}, q_{ADVP})$ $\quad p_{JJ} \rightarrow \mathrm{JJ}(q_c)$

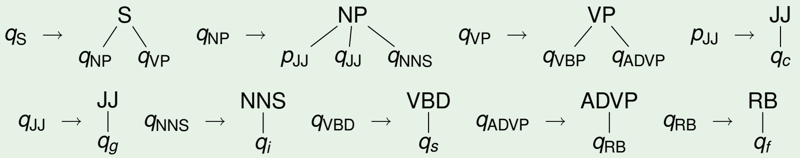
$q_{JJ} \rightarrow \mathrm{JJ}(q_g)$ $\quad q_{NNS} \rightarrow \mathrm{NNS}(q_i)$ $\quad q_{VBD} \rightarrow \mathrm{VBD}(q_s)$ $\quad q_{ADVP} \rightarrow \mathrm{ADVP}(q_{RB})$ $\quad q_{RB} \rightarrow \mathrm{RB}(q_f)$

Derivation

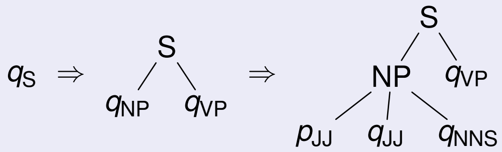
$$q_S \Rightarrow \mathrm{S}(q_{NP}, q_{VP})$$

Regular tree language

Example



Derivation



Regular tree languages

Principal properties

- Finite languages are regular
- Closed under all Boolean operations
- Closed under relabelings, linear homomorphisms, inverse homomorphisms
- Can be determinized and minimized (bottom-up)

Summary

They are basically the tree version of finite-state automata with the same nice properties.

Trees and their yield

Definition

The yield of a tree is the string of its leaves (in natural order).

Theorem

- *The yield language of a regular tree language is context-free.*
- *Each context-free language is the yield of a regular tree language.*

Questions

Question

Is $\{\sigma(t, t) \mid t \text{ arbitrary tree}\}$ regular?

Questions

Question

Is $\{\sigma(t, t) \mid t \text{ arbitrary tree}\}$ regular?

Answer

NO!

Questions

Question

Is $\{\sigma(t, t) \mid t \text{ arbitrary tree}\}$ regular?

Answer

NO!

Question

Is $\{\sigma(\gamma^n(\alpha), \gamma^n(\alpha)) \mid n \in \mathbb{N}\}$ regular?

Questions

Question

Is $\{\sigma(t,t) \mid t \text{ arbitrary tree}\}$ regular?

Answer

NO!

Question

Is $\{\sigma(\gamma^n(\alpha), \gamma^n(\alpha)) \mid n \in \mathbb{N}\}$ regular?

Answer

NO!

Remark

Not every tree language with context-free yield language is regular!

Back to parsing

Observation

Most CFG-parsers are regular tree grammars ($+$ control) because

- they are based on a CFG ($\rightarrow$ local tree grammar) and
- have hidden states (or features)

Alternative

The features can be made explicit in the parse tree structure.

Back to parsing

Observation

Most CFG-parsers are regular tree grammars ($+$ control) because

- they are based on a CFG ($\rightarrow$ local tree grammar) and
- have hidden states (or features)

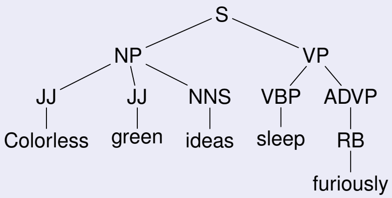
Alternative

The features can be made explicit in the parse tree structure.

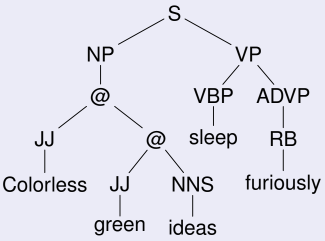
Contents

Binarization

Input tree



Binarized tree

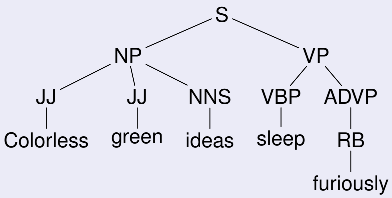


Theorem

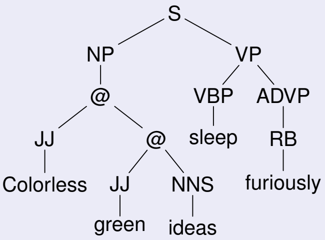
A tree language is regular if and only if its binarization is regular

Binarization

Input tree



Binarized tree



Theorem

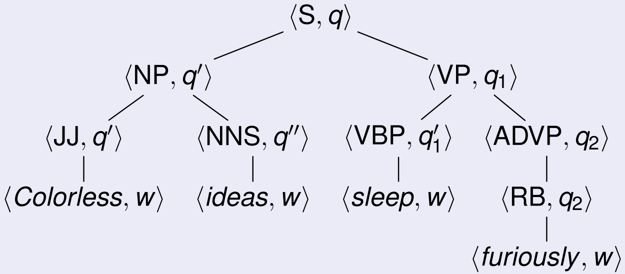
A tree language is regular if and only if its binarization is regular

Individual runs

Run on the yield

(p) Colorless (p_1) ideas (p_2) sleep (p_3) furiously (p')

Run on the input tree

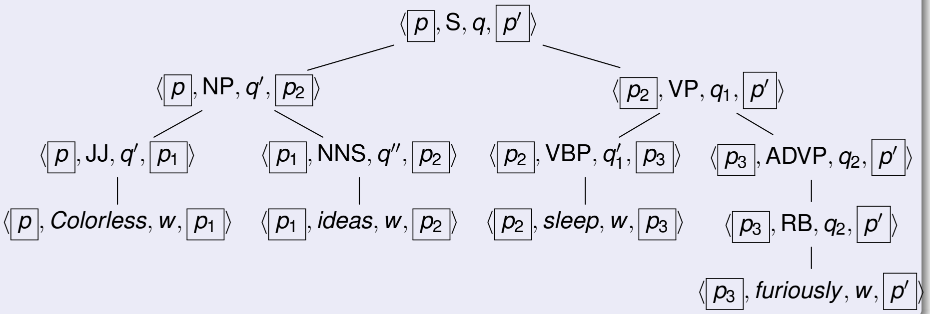


Bar-Hillel construction

Run on the yield

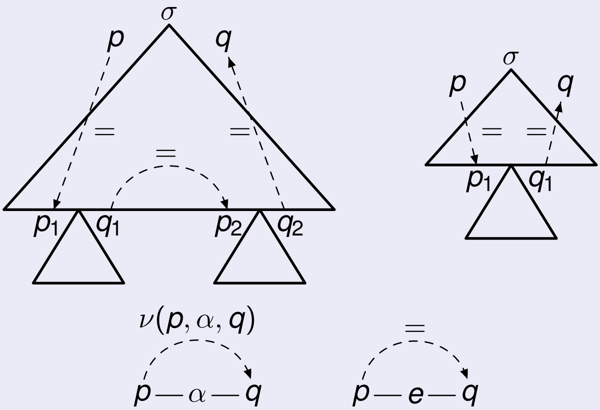
(p) Colorless (p_1) ideas (p_2) sleep (p_3) furiously (p')

Composite run



Bar-Hillel construction (cont'd)

Illustration



Bar-Hillel construction (cont'd)

Theorem

The regular restriction of a regular tree language is regular

Remark

Complexity: $O(mn^3)$

- m: size of the regular tree grammar
- n: size of the regular grammar (or input string)

Conclusion

We can parse with regular tree grammars in $O(mn^3)$.

Contents

Top-down Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

$$\langle \boxed{1}, \mathrm{S}, q, \boxed{5} \rangle$$

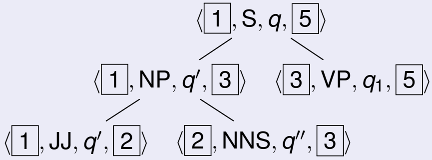
$$\langle \boxed{1}, \mathrm{NP}, q', \boxed{3} \rangle \qquad \langle \boxed{3}, \mathrm{VP}, q_1, \boxed{5} \rangle$$

Top-down Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

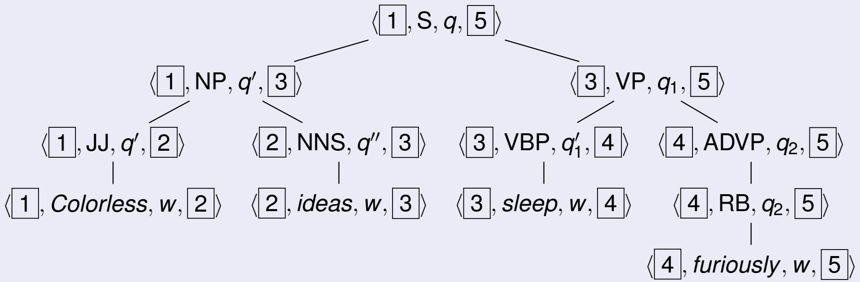


Top-down Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run



Bottom-up Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

$$\langle \boxed{1}, \textit{Colorless}, w, \boxed{2} \rangle$$

Bottom-up Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

$$\langle \boxed{1}, \text{JJ}, q', \boxed{2} \rangle$$

$$|$$

$$\langle \boxed{1}, \textit{Colorless}, w, \boxed{2} \rangle$$

Bottom-up Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

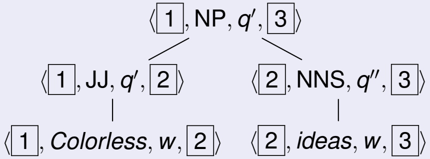
$$\begin{array}{c} \langle \boxed{1}, \mathrm{JJ}, q', \boxed{2} \rangle \\ | \\ \langle \boxed{1}, \textit{Colorless}, w, \boxed{2} \rangle \end{array} \qquad \begin{array}{c} \langle \boxed{2}, \mathrm{NNS}, q'', \boxed{3} \rangle \\ | \\ \langle \boxed{2}, \textit{ideas}, w, \boxed{3} \rangle \end{array}$$

Bottom-up Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run

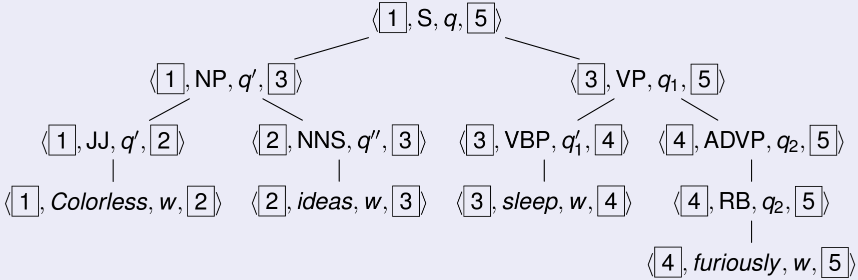


Bottom-up Bar-Hillel construction

Run on the yield

(1) Colorless (2) ideas (3) sleep (4) furiously (5)

Composite run



Summary

Key points

- regular tree grammar as efficient tree data structure
- context-free behavior
- basis for most syntax-based translation models

Further models

- weaker models are generally inadequate
- tree adjoining grammars (more expressive, but worse computational properties)
- Automata on directed acyclic graphs (see Daniel's lecture)

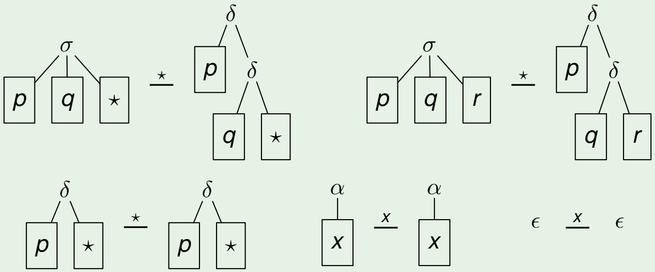
Contents

Extended Top-down Tree Transducer

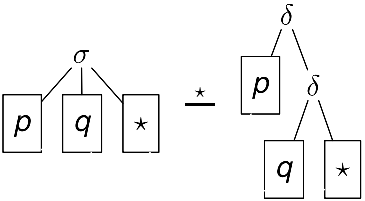
Definition

Each rule now has an input and an output side, which are both full trees (not just a single symbol followed by states)

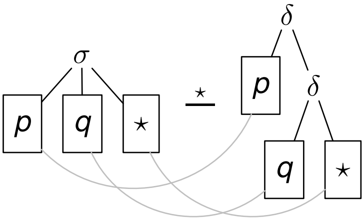
Example



Link Structure

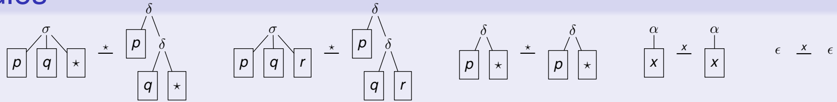


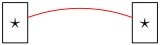
Link Structure



Derivation

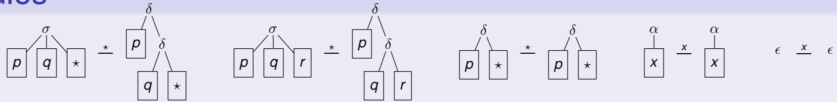
Rules

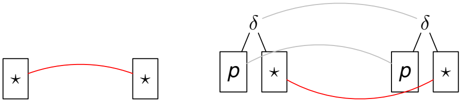




Derivation

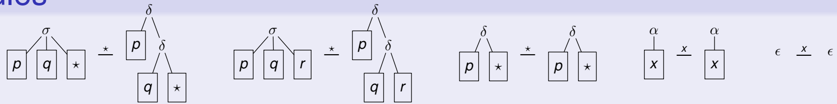
Rules

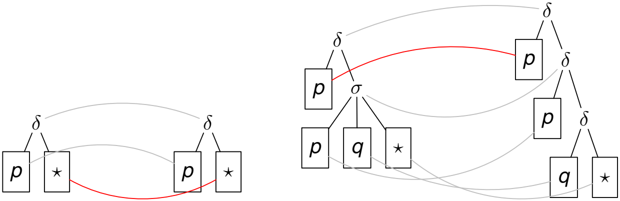




Derivation

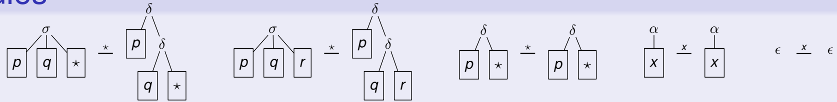
Rules

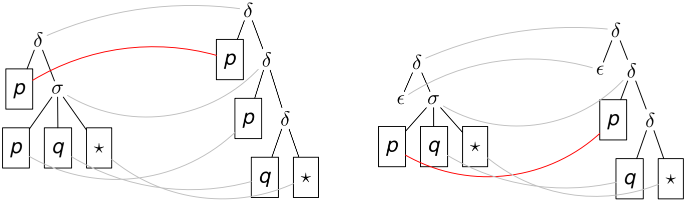




Derivation

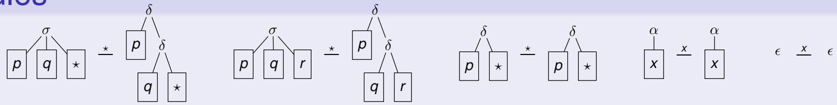
Rules

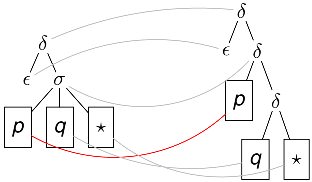


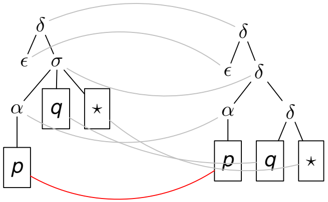


Derivation

Rules

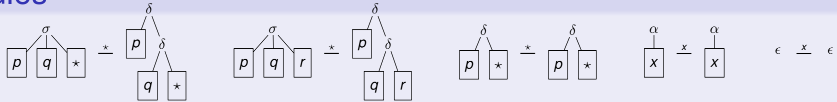


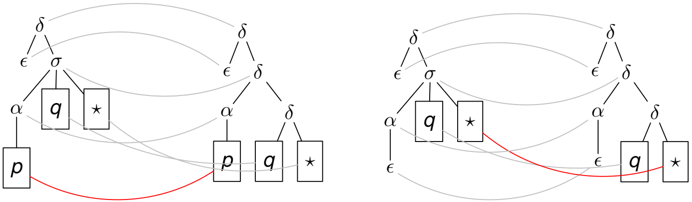




Derivation

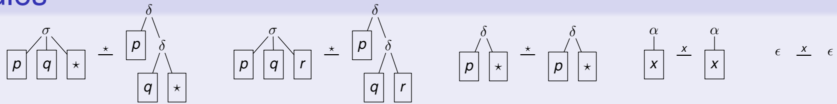
Rules

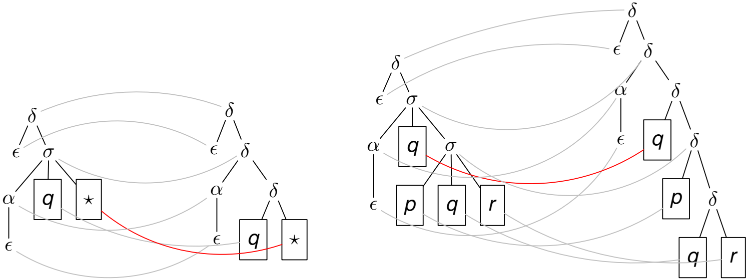




Derivation

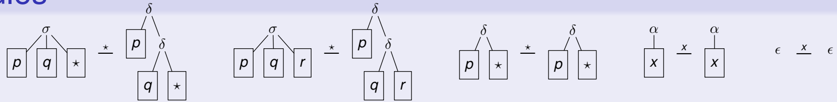
Rules

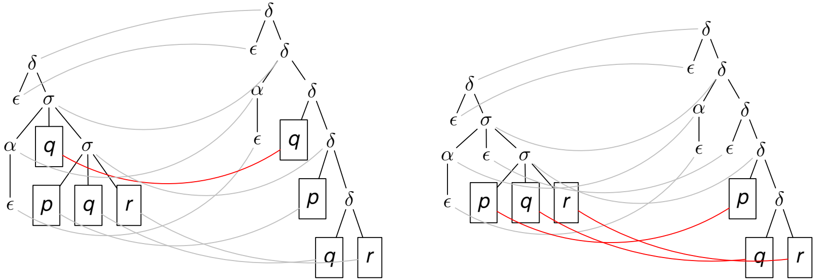




Derivation

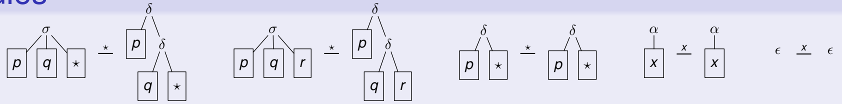
Rules

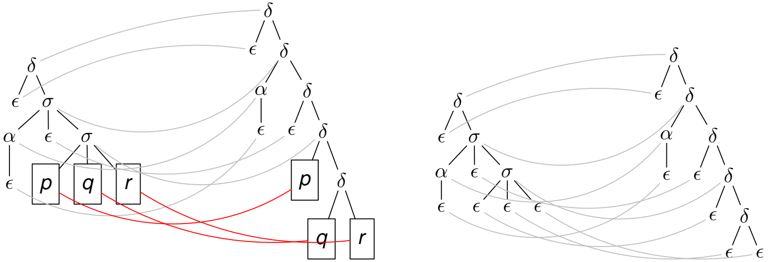




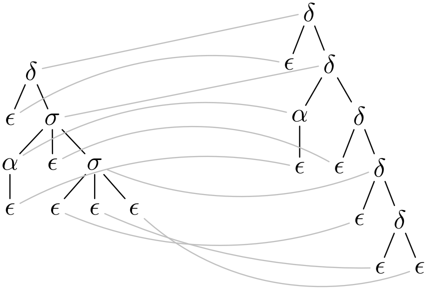
Derivation

Rules

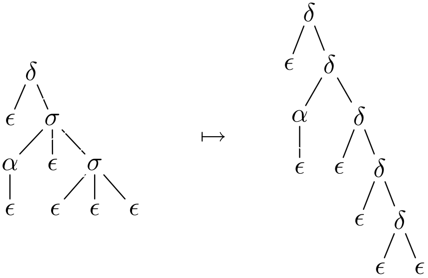




Semantics of XTOP



Semantics of XTOP



References

- Bar-Hillel, Perles, Shamir: *On formal properties of simple phrase structure grammars.* Language and Information, 1964
- Berstel, Reutenauer: Recognizable formal power series on trees. *Theoret. Comput. Sci.* 18, p. 115–148, 1982
- Borchardt: The theory of recognizable tree series. Ph.D. thesis TU Dresden, 2005
- Gécseg, Steinby: *Tree Automata*. Akadémiai Kiadó, Budapest 1984
- Nederhof, Satta: Probabilistic parsing as intersection. In IWPT 2003

Thank you for your attention!