Tree Transducers in Machine Translation

Andreas Maletti

Universität Stuttgart

andreas.maletti@ims.uni-stuttgart.de

Stuttgart — March 2, 2011

Machine translation

Applications

- Technical manuals

Example (An mp3 player)

The synchronous manifestation of lyrics is a procedure for can broadcasting the music, waiting the mp3 file at the same time showing the lyrics.

With the this kind method that the equipments that synchronous function of support up broadcast to make use of document create setup, you can pass the LCD window way the check at the document contents that broadcast.

That procedure returns offerings to have to modify, and delete, and stick top , keep etc. edit function.

Machine translation

Applications

- Technical manuals

Example (An mp3 player)

The synchronous manifestation of lyrics is a procedure for can broadcasting the music, waiting the mp3 file at the same time showing the lyrics.

With the this kind method that the equipments that synchronous function of support up broadcast to make use of document create setup, you can pass the LCD window way the check at the document contents that broadcast.

That procedure returns offerings to have to modify, and delete, and stick top , keep etc. edit function.

Machine translation

Applications

- Technical manuals

Example (An mp3 player)

The synchronous manifestation of lyrics is a procedure for can broadcasting the music, waiting the mp3 file at the same time showing the lyrics.
With the this kind method that the equipments that synchronous function of support up broadcast to make use of document create setup, you can pass the LCD window way the check at the document contents that broadcast.
That procedure returns offerings to have to modify, and delete, and stick top , keep etc. edit function.

Machine translation

Applications

- Technical manuals
- TripAdvisor®

Example (Hotel Uppsala, Sweden)

Wir hatten die Zimmer eingestuft wird als "Superior" weil sie renoviert wurde im letzten Jahr oder zwei. Unsere Zimmer hatten Parkettboden und waren sehr geräumig. Man musste allerdings nicht musste seitwärts bewegen.

Machine translation

Applications

- Technical manuals
- TripAdvisor®

Example (Hotel Uppsala, Sweden)

Nos alojamos en habitaciones clasificado como “superior” porque se lo habían renovado en el año pasado o dos. Nuestras habitaciones tenían suelos de madera y eran espaciosas. No te tenías que caminar arriba para movernos por allí.

Machine translation

Applications

- Technical manuals
- TripAdvisor®

Example (Hotel Uppsala, Sweden)

Wir hatten die Zimmer eingestuft wird als "Superior" weil sie renoviert wurde im letzten Jahr oder zwei. Unsere Zimmer hatten Parkettboden und waren sehr geräumig. Man musste allerdings nicht musste seitwärts bewegen.

— *We stayed in rooms classified as "superior" because they had been renovated in the last year or two. Our rooms had wood floors and were roomy. You didn't have to walk sideways to move around.*

Machine translation

Applications

- Technical manuals
- TripAdvisor®
- Military

Example (Jones, Shen, Herzog 2009)

Soldier:	Okay, what is your name?
Local:	Abdul.
Soldier:	And your last name?
Local:	Al Farran.

Machine translation

Applications

- Technical manuals
- TripAdvisor®
- Military

Example (Jones, Shen, Herzog 2009)

Soldier:	Okay, what is your name?
Local:	Abdul.
Soldier:	And your last name?
Local:	Al Farran.

Speech-to-text machine translation

Soldier:	Okay, what's your name?
Local:	milk a mechanic and I am here I mean yes

Machine translation

Applications

- Technical manuals
- TripAdvisor®
- Military

Example (Jones, Shen, Herzog 2009)

Soldier:	Okay, what is your name?
Local:	Abdul.
Soldier:	And your last name?
Local:	Al Farran.

Speech-to-text machine translation

Soldier:	Okay, what's your name?
Local:	milk a mechanic and I am here I mean yes
Soldier:	What is your last name?
Local:	every two weeks my son's name is ismail

Machine translation

Applications

- Technical manuals
- TripAdvisor®
- Military
- MSDN, Knowledge Base
- ...

Machine translation (cont'd)

Systems

- GOOGLE translate `translate.google.com`
- BING translator `www.microsofttranslator.com`
- LANGUAGE WEAVER + SDL `www.freetranslation.com`
- ...

Machine translation (cont'd)

Systems

- GOOGLE translate — `translate.google.com`
- BING translator — `www.microsofttranslator.com`
- LANGUAGE WEAVER + SDL — `www.freetranslation.com`
- ...

Try them!

Machine translation (cont'd)

History

1. Dark age (60s–90s)
   - rule-based systems (e.g., SYSTRAN)
   - CHOMSKYAN approach
   - perfect translation, poor coverage
2. Reformation (1991–present)
   - word-based, phrase-based, syntax-based systems
   - statistical approach
   - cheap, automatically trained
3. Potential future
   - semantics-based systems (e.g., FRAMENET)
   - semi-supervised, statistical approach
   - basic understanding of translated text

Machine translation (cont'd)

History

1. Dark age (60s–90s)
   - rule-based systems (e.g., SYSTRAN)
   - CHOMSKYAN approach
   - perfect translation, poor coverage
2. Reformation (1991–present)
   - word-based, phrase-based, syntax-based systems
   - statistical approach
   - cheap, automatically trained
3. Potential future
   - semantics-based systems (e.g., FRAMENET)
   - semi-supervised, statistical approach
   - basic understanding of translated text

Machine translation (cont'd)

History

1. Dark age (60s–90s)
   - rule-based systems (e.g., SYSTRAN)
   - CHOMSKYAN approach
   - perfect translation, poor coverage
2. Reformation (1991–present)
   - word-based, phrase-based, syntax-based systems
   - statistical approach
   - cheap, automatically trained
3. Potential future
   - semantics-based systems (e.g., FRAMENET)
   - semi-supervised, statistical approach
   - basic understanding of translated text

Machine translation (cont'd)

Schema

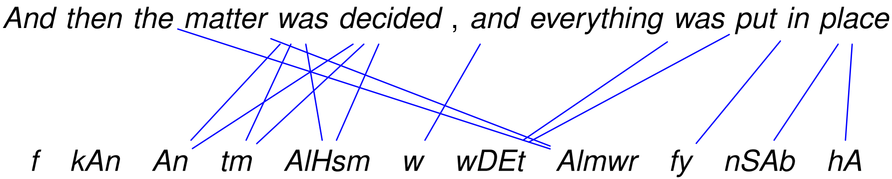
Input $\longrightarrow$ Machine translation system $\longrightarrow$ Language model $\longrightarrow$ Output

Machine translation (cont'd)

Schema

Input $\longrightarrow$ Machine translation system $\longrightarrow$ Language model $\longrightarrow$ Output

Word-based system (FST)



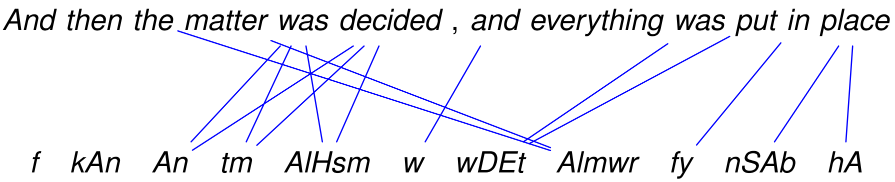
Derivation

Input:

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

Output:

Word-based system (FST)



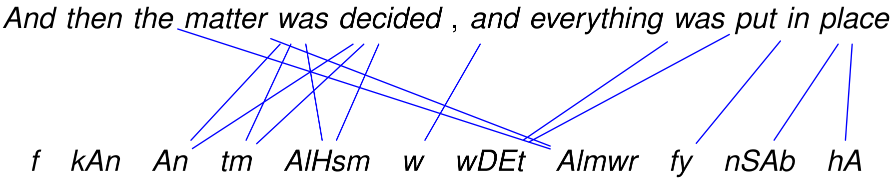
Derivation

Input:

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

Output:

Word-based system (FST)



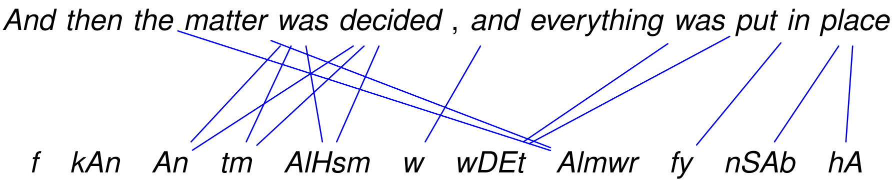
Derivation

Input:

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

Output:

Word-based system (FST)



Derivation

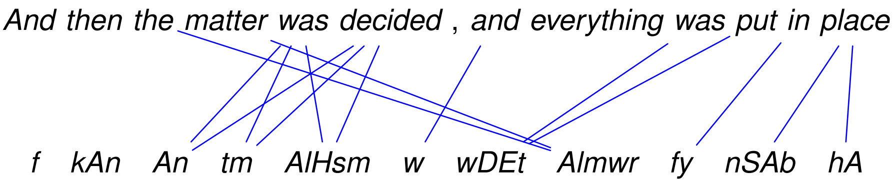
Input:

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

Output:

f

Word-based system (FST)



Derivation

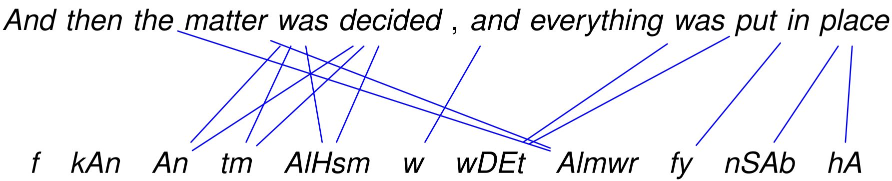
Input:

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

Output:

f kAn

Word-based system (FST)



Derivation

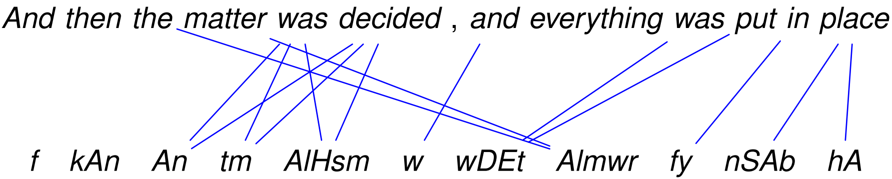
Input:

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

Output:

f kAn

Word-based system (FST)



Derivation

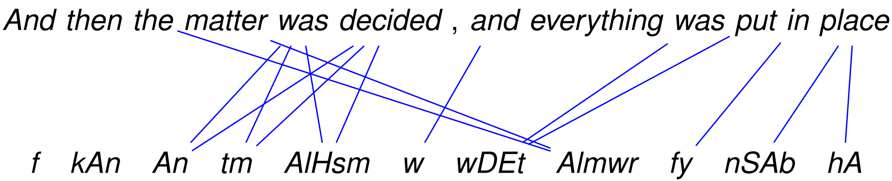
Input:

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

Output:

f kAn

Word-based system (FST)



Derivation

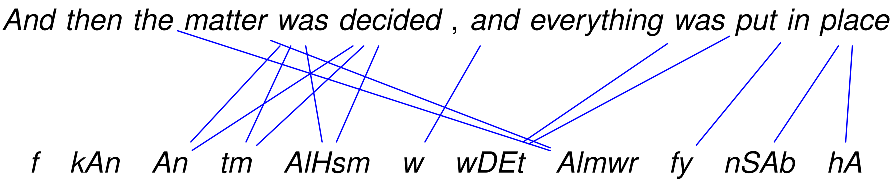
Input:

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

Output:

f kAn

Word-based system (FST)



Derivation

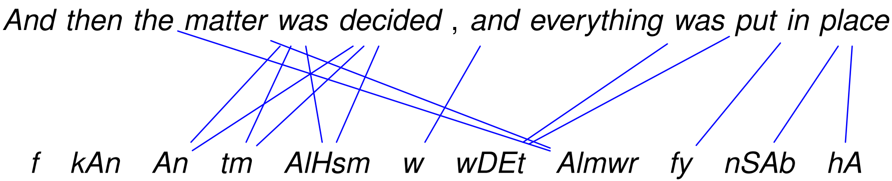
Input:

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

Output:

f kAn An tm AlHsm

Word-based system (FST)



Derivation

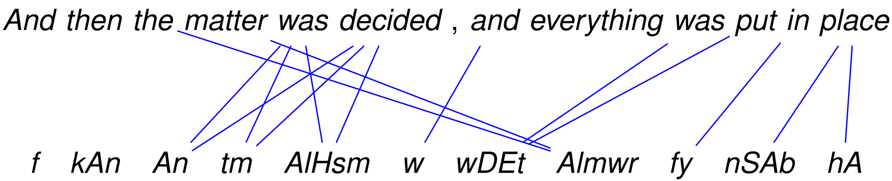
Input:

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

Output:

f kAn An tm AlHsm

Word-based system (FST)



Derivation

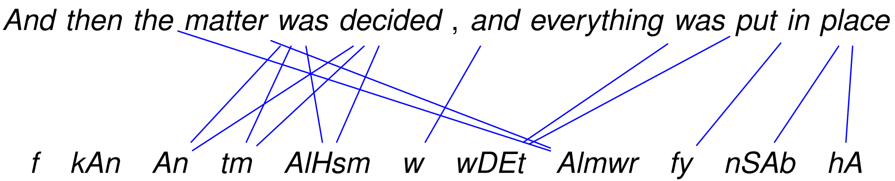
Input:

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

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

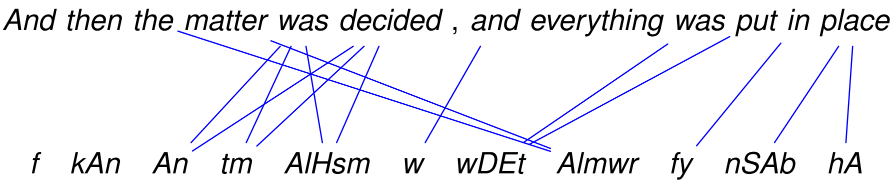
Input:

the matter | *was* | *put in place*

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

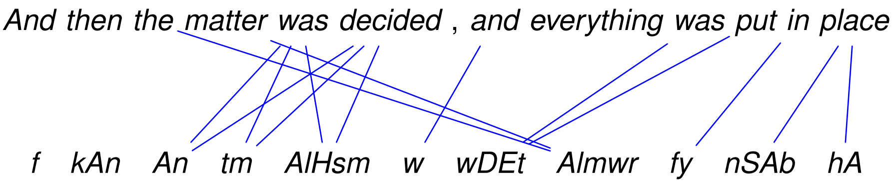
Input:

the matter was *put* *in place*

Output:

f kAn An tm AlHsm w

Word-based system (FST)



Derivation

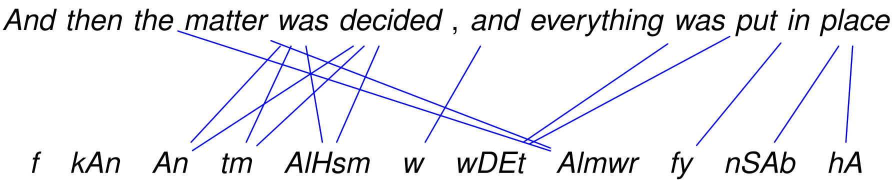
Input:

the matter *in* *place*

Output:

f kAn An tm AlHsm w wDEt

Word-based system (FST)



Derivation

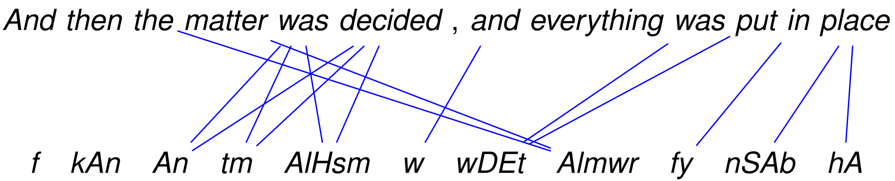
Input:

in *place*

Output:

f kAn An tm AlHsm w wDEt Almwr

Word-based system (FST)



Derivation

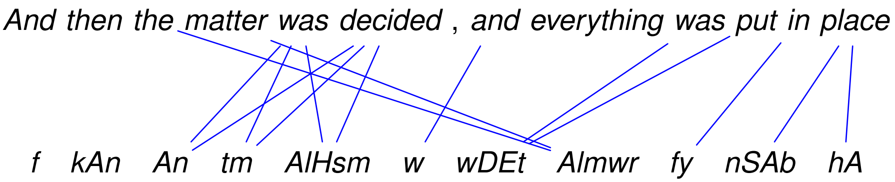
Input:

place

Output:

f kAn An tm AlHsm w wDEt Almwr fy

Word-based system (FST)



Derivation

Input:

Output:

f kAn An tm AlHsm w wDEt Almwr fy nSAb hA

Phrase-based machine translation

Schema

Input $\longrightarrow$ Machine translation system $\longrightarrow$ Language model $\longrightarrow$ Output

Phrase-based systems

Input $\longrightarrow$ Segmenter $\longrightarrow$ Machine translation system $\longrightarrow$ Language model $\longrightarrow$ Output

Phrase-based machine translation

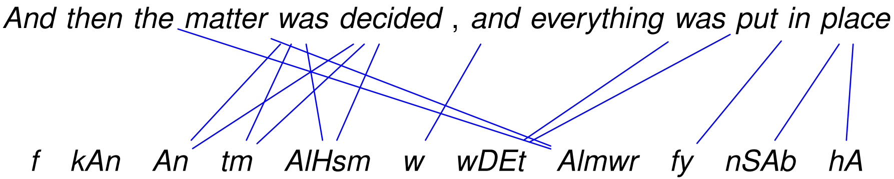
Schema

Input ⟶ [Machine translation system] ⟶ [Language model] ⟶ Output

Phrase-based systems

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

Phrase-based system (FST+Perm)



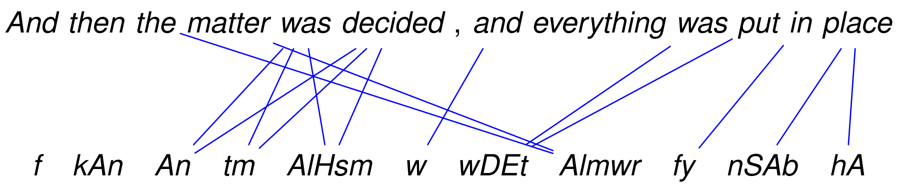
Derivation

Input:

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

Output:

Phrase-based system (FST+Perm)



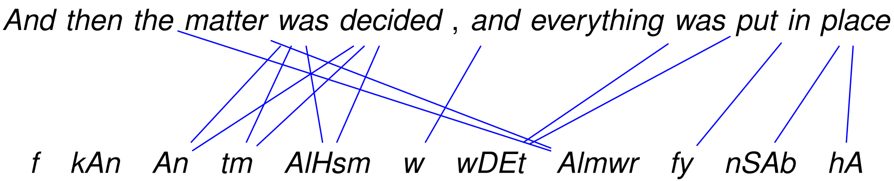
Derivation

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)

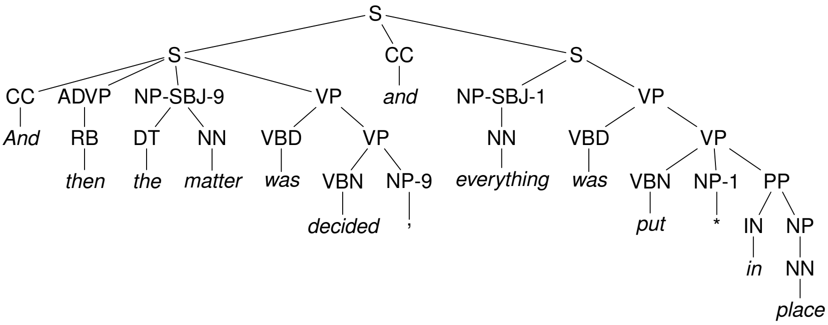
Phrase-based systems

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

Syntax-based systems

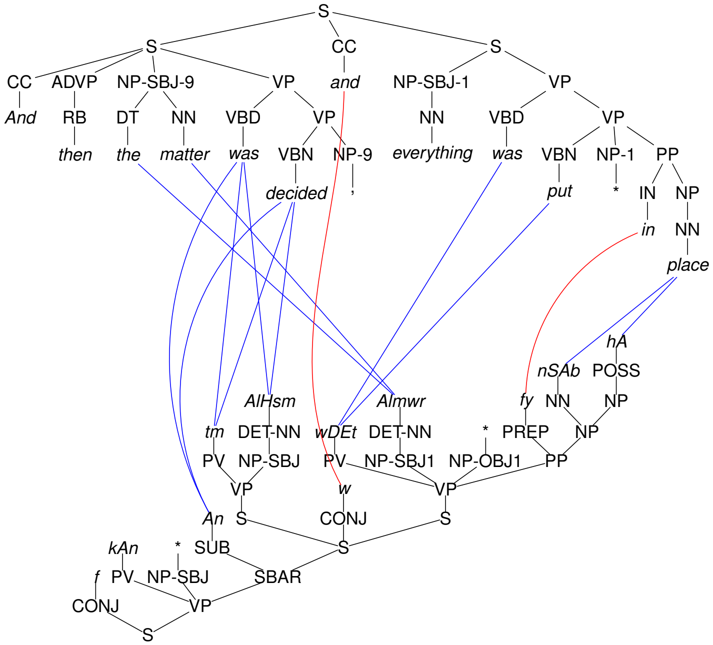
Input ⟶ Parser ⟶ Machine translation system ⟶ Language model ⟶ Output

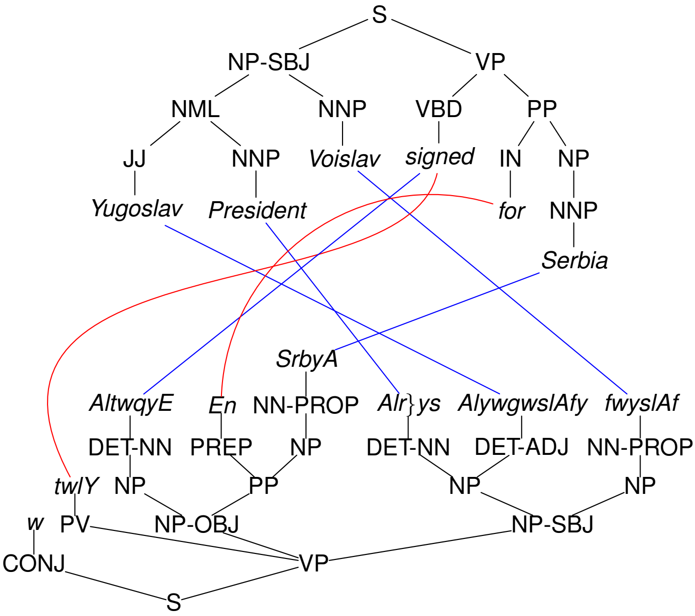
Parser



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

(thanks to KEVIN KNIGHT for the data)


S
S
CC
S
CC
ADVP
NP-SBJ-9
VP
and
NP-SBJ-1
VP
And
RB
DT
NN
VBD
VP
NN
VBD
VP
then
the
matter
was
VBN
NP-9
everything
was
VBN
NP-1
PP
decided
,
put
*
IN
NP
in
NN
place
hA
nSAb
POSS
AlHsm
Almwr
fy
NN
NP
tm
DET-NN
wDEt
DET-NN
*
PREP
NP
PV
NP-SBJ
PV
NP-SBJ1
NP-OBJ1
PP
VP
w
VP
An
S
CONJ
S
kAn
*
SUB
S
f
PV
NP-SBJ
SBAR
CONJ
VP
S


S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

Contents

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Example

- BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$
- Semiring $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$ of probabilities
- Tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- Any field, ring, etc.

Most of the talk: BOOLEAN semiring

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Example

- BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$
- Semiring $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$ of probabilities
- Tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- Any field, ring, etc.

Most of the talk: BOOLEAN semiring

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Example

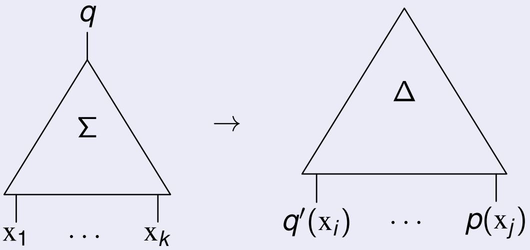
- BOOLEAN semiring $(\{0, 1\}, \max, \min, 0, 1)$
- Semiring $(\mathbb{R}_{\geq 0}, +, \cdot, 0, 1)$ of probabilities
- Tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$
- Any field, ring, etc.

Most of the talk: BOOLEAN semiring

Syntax

Definition (Arnold, Dauchet 1976, Graehl, Knight 2004)

Extended top-down tree transducer (XTOP) $M = (Q, \Sigma, \Delta, I, R)$ with finitely many rules

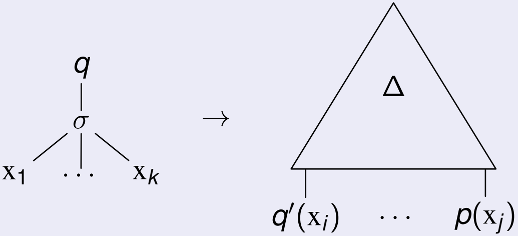


- $q, q', p \in Q$ are states
- $i, j \in \{1, \dots, k\}$

Syntax (cont'd)

Definition (Rounds 1970, Thatcher 1970)

- Top-down tree transducer (TOP) if all rules

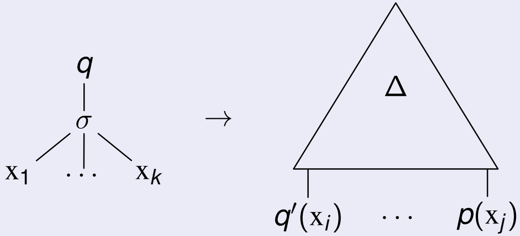


- linear if no variable occurs twice in r for all rules $l \to r$
- nondeleting if $\text{var}(l) = \text{var}(r)$ for all rules $l \to r$

Syntax (cont'd)

Definition (Rounds 1970, Thatcher 1970)

- Top-down tree transducer (TOP) if all rules

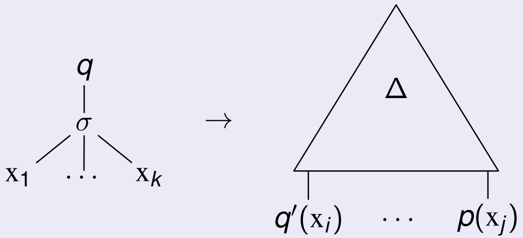


- linear if no variable occurs twice in r for all rules $l \to r$
- nondeleting if $\text{var}(l) = \text{var}(r)$ for all rules $l \to r$

Syntax (cont'd)

Definition (Rounds 1970, Thatcher 1970)

- Top-down tree transducer (TOP) if all rules

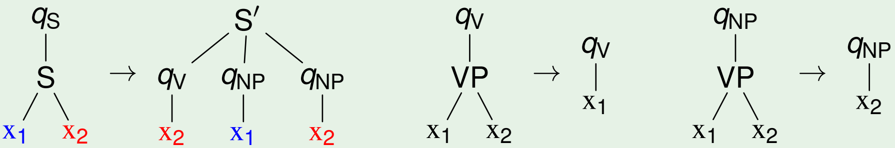


- linear if no variable occurs twice in r for all rules $l \to r$
- nondeleting if $\mathrm{var}(l) = \mathrm{var}(r)$ for all rules $l \to r$

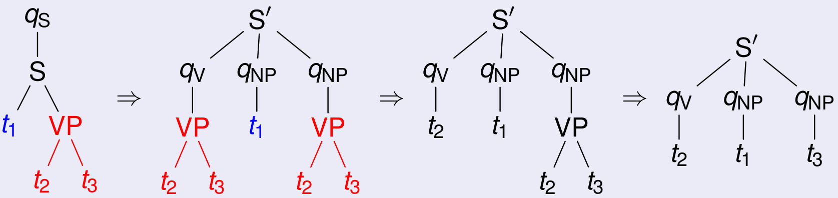
Semantics

Example

States $\{q_S, q_V, q_{NP}\}$ of which only q_S is initial



Derivation

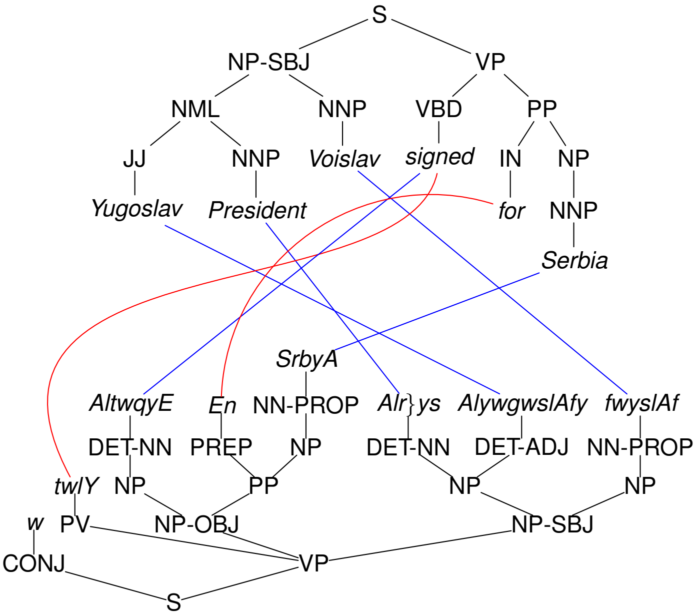


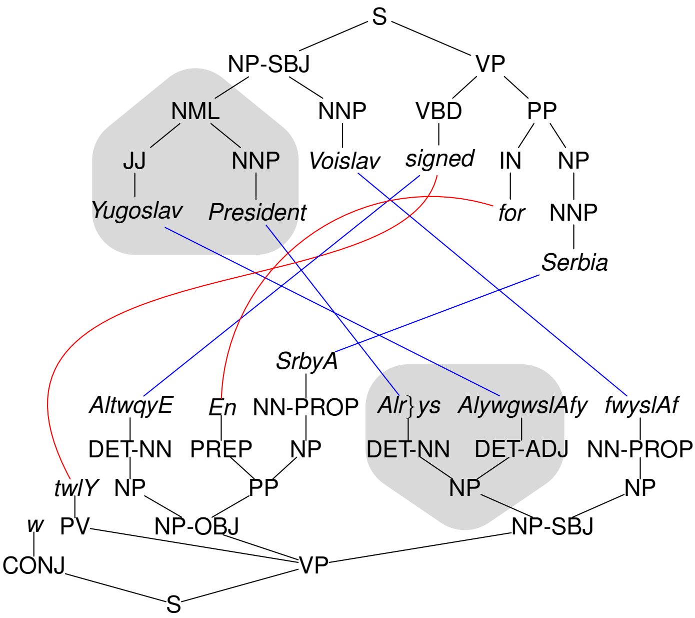
Semantics (cont'd)

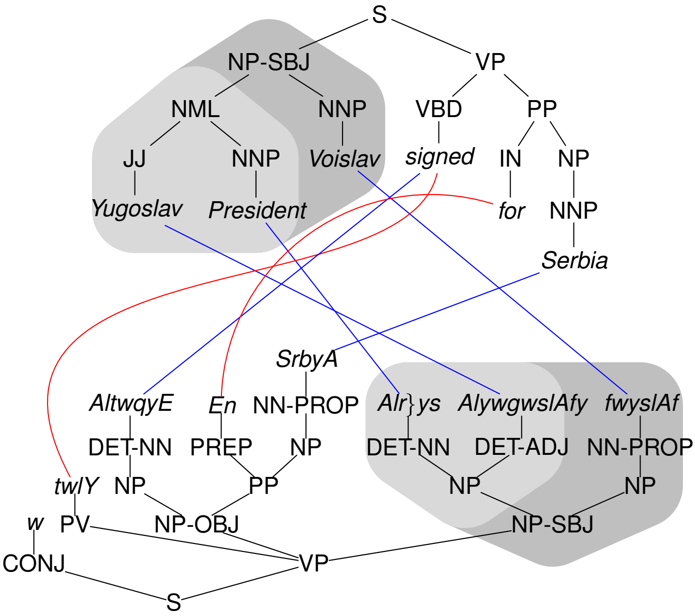
Definition

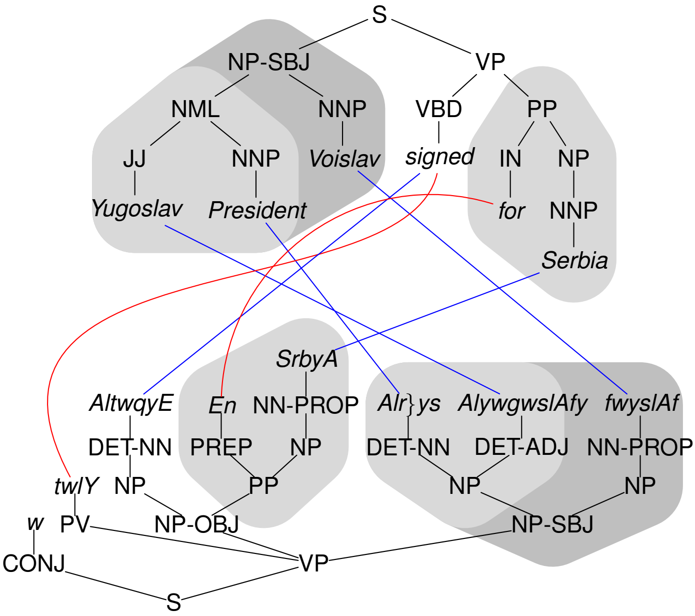
Computed transformation:

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in I \colon q(t) \Rightarrow^* u\}$$

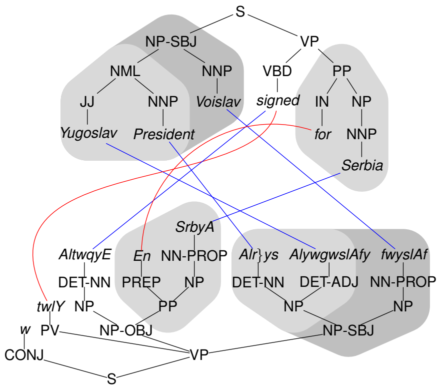

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

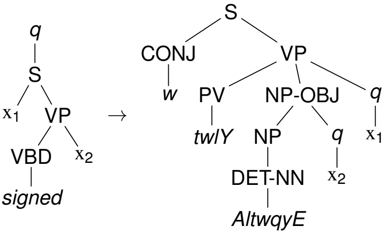

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S


S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

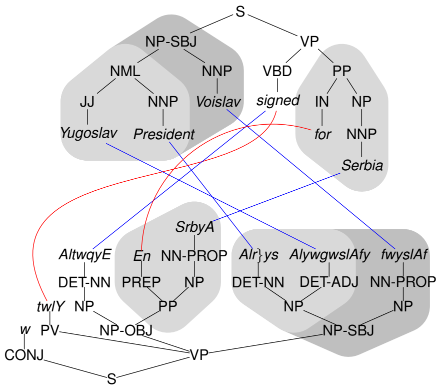

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

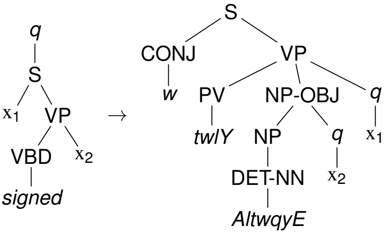
Rule extraction

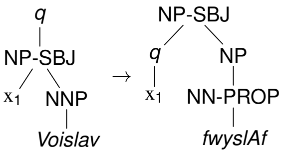




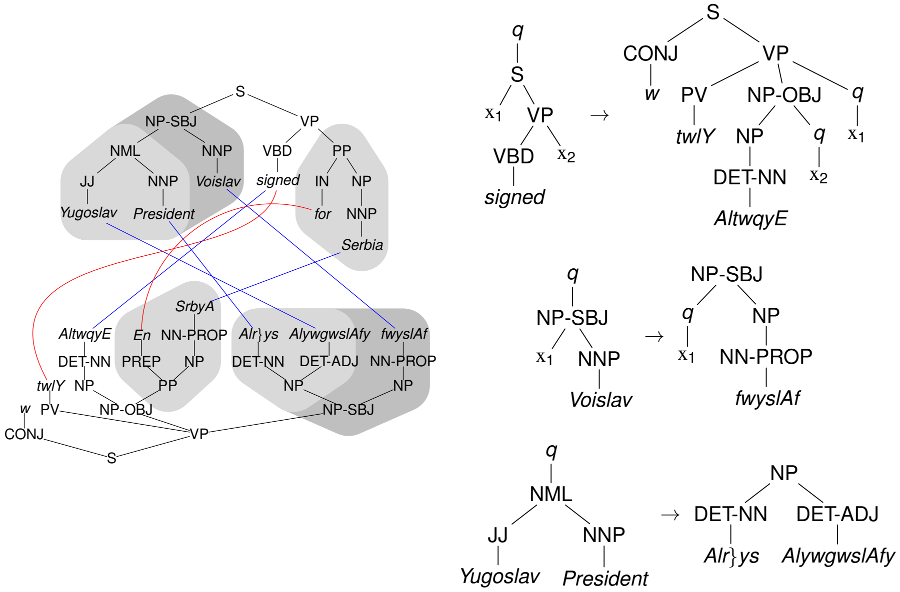
Rule extraction





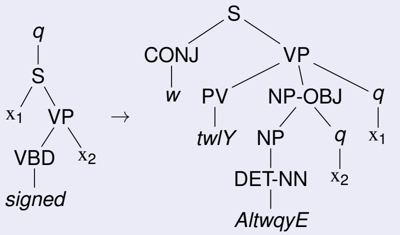


Rule extraction



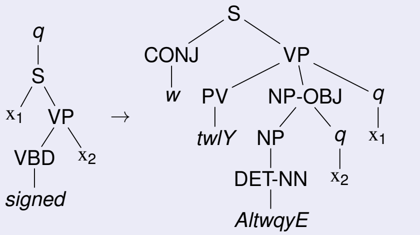
Symmetry

Original rule

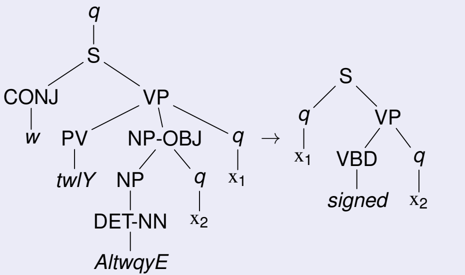


Symmetry

Original rule

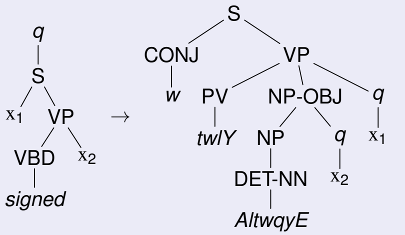


Inverted rule

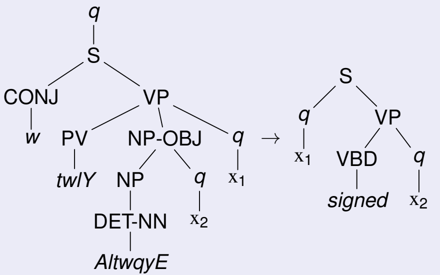


Symmetry

Original rule



Inverted rule



Linear nondeleting XTT can be inverted

Preservation of regularity

Schematics

Input $\longrightarrow$ Parser $\longrightarrow$ XTT $\longrightarrow$ Language model $\longrightarrow$ Output

Parse trees

- best parse tree
- *n*-best parses
- all parses

Can all be represented by regular tree language

Preservation of regularity

Schematics

Input ⟶ Parser ⟶ Parse trees ⟶ XTT ⟶ ...

Parse trees

- best parse tree
- n-best parses
- all parses

Can all be represented by regular tree language

Preservation of regularity

Schematics

Input ⟶ Parser ⟶ Parse trees ⟶ XTT ⟶ …

Parse trees

- best parse tree
- *n*-best parses
- all parses

Can all be represented by regular tree language

Preservation of regularity

Schematics

Input $\longrightarrow$ Parser $\longrightarrow$ Parse trees $\longrightarrow$ XTT $\longrightarrow \dots$

Parse trees

- best parse tree
- *n*-best parses
- all parses

Can all be represented by regular tree language

Preservation of regularity (cont'd)

Schematics

Regular language $\longrightarrow$ XTT $\longrightarrow$ Regular language?

Approach

- Input restriction
- Project to output

Result

Linear XTT preserve regularity

Preservation of regularity (cont'd)

Schematics

Regular language ⟶ XTT ⟶ Regular language?

Approach

- Input restriction
- Project to output

Result

Linear XTT preserve regularity

Preservation of regularity (cont'd)

Schematics

Regular language $\longrightarrow$ XTT $\longrightarrow$ Regular language?

Approach

- Input restriction
- Project to output

Result

Linear XTT preserve regularity

Composition

Schematics

Parse trees $\longrightarrow$ XTT $\longrightarrow$ ...

Example (YAMADA, KNIGHT 2002)

- Reorder
- Insert words
- Translate words

Composition

Schematics

Parse trees $\longrightarrow$ [Stage 1: XTT] $\longrightarrow$ [Stage 2: XTT] $\longrightarrow$ [Stage 3: XTT] $\longrightarrow \ldots$

Example (Yamada, Knight 2002)

- Reorder
- Insert words
- Translate words

Composition

Schematics

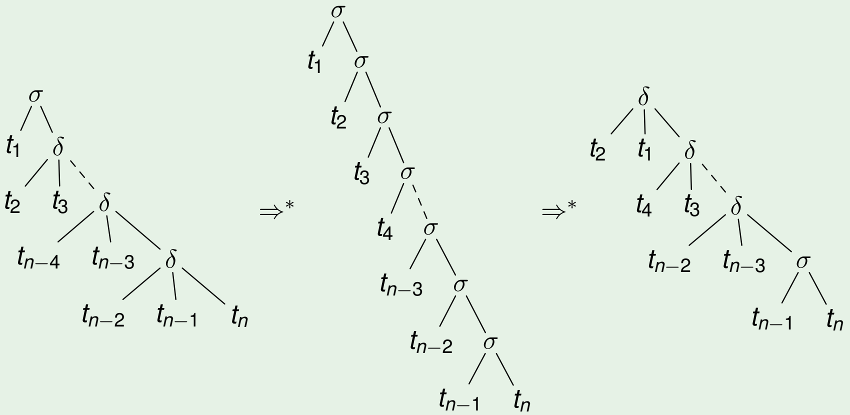
Parse trees $\longrightarrow$ [Composed XTT] $\longrightarrow$ . . .

Example (YAMADA, KNIGHT 2002)

- Reorder
- Insert words
- Translate words

Composition (cont'd)

Example (ARNOLD, DAUCHET 1982)



Summary

Model \ Criterion	EXPR	SYM	PRES	PRES^{-1}	COMP
Linear nondeleting TOP	✗	✗	✓	✓	✓
Linear TOP	✗	✗	✓	✓	✗
Linear TOP$^{\mathrm{R}}$	✗	✗	✓	✓	✓
General TOP	✗	✗	✗	✓	✗
General TOP$^{\mathrm{R}}$	✓	✗	✗	✓	✗
Linear nondeleting XTOP	✓	✓	✓	✓	✗
Linear XTOP	✓	✗	✓	✓	✗
Linear XTOP$^{\mathrm{R}}$	✓	✗	✓	✓	✗
General XTOP	✓	✗	✗	✓	✗
General XTOP$^{\mathrm{R}}$	✓	✗	✗	✓	✗

Summary

Model \ Criterion	EXPR	SYM	PRES	PRES^{-1}	COMP
Linear nondeleting TOP	✗	✗	✓	✓	✓
Linear TOP	✗	✗	✓	✓	✗
Linear TOP$^{\mathrm{R}}$	✗	✗	✓	✓	✓
General TOP	✗	✗	✗	✓	✗
General TOP$^{\mathrm{R}}$	✓	✗	✗	✓	✗
Linear nondeleting XTOP	✓	✓	✓	✓	✗
Linear XTOP	✓	✗	✓	✓	✗
Linear XTOP$^{\mathrm{R}}$	✓	✗	✓	✓	✗
General XTOP	✓	✗	✗	✓	✗
General XTOP$^{\mathrm{R}}$	✓	✗	✗	✓	✗
Comp. closure ln-XTOP	✓	✓	✓	✓	✓
“composable” ln-XTOP	?	?	✓	✓	✓

Implementation

TIBURON [MAY, KNIGHT 2006]

- Implements XTOP (and tree automata; everything also weighted)
- Framework with command-line interface
- Optimized for machine translation

Algorithms

- Application of XTOP to input tree/language
- Backward application of XTOP to output language
- Composition (for some XTOP)

Example

```
qNP.NP(DT(the) N(boy)) -> NP(N(atefl))
```

Implementation

TIBURON [MAY, KNIGHT 2006]

- Implements XTOP (and tree automata; everything also weighted)
- Framework with command-line interface
- Optimized for machine translation

Algorithms

- Application of XTOP to input tree/language
- Backward application of XTOP to output language
- Composition (for some XTOP)

Example

```
qNP.NP(DT(the) N(boy)) -> NP(N(atefl))
```

Implementation

TIBURON [MAY, KNIGHT 2006]

- Implements XTOP (and tree automata; everything also weighted)
- Framework with command-line interface
- Optimized for machine translation

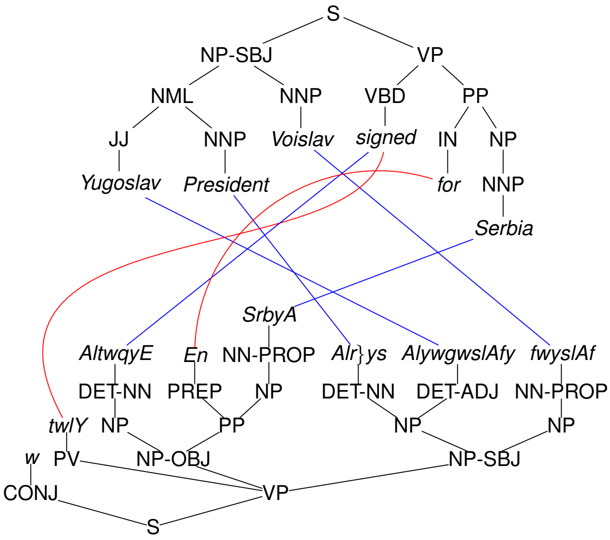
Algorithms

- Application of XTOP to input tree/language
- Backward application of XTOP to output language
- Composition (for some XTOP)

Example

```
qNP.NP(DT(the) N(boy)) -> NP(N(atefl))
```

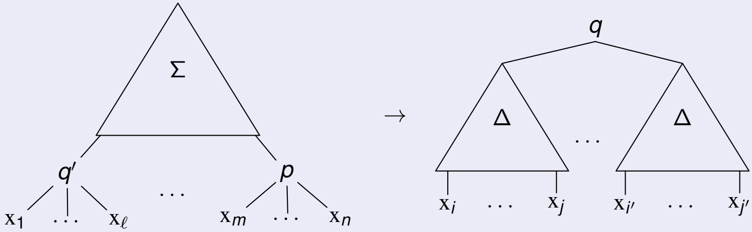
Multi Bottom-up Tree Transducers



Syntax

Definition

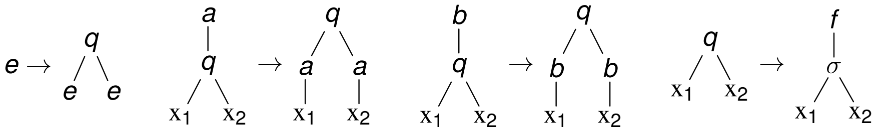
Extended multi bottom-up tree transducer (XMBOT)
is $M = (Q, \Sigma, \Delta, F, R)$ with finitely many rules



- $q', p, q \in Q$ are now ranked *states*
- $F \subseteq Q_1$ final states

Example

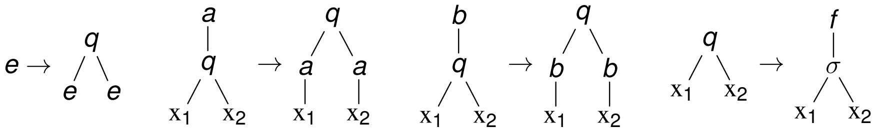
States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules



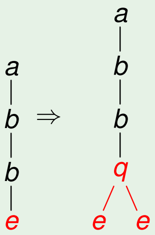
Example (Derivation)

Example

States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules

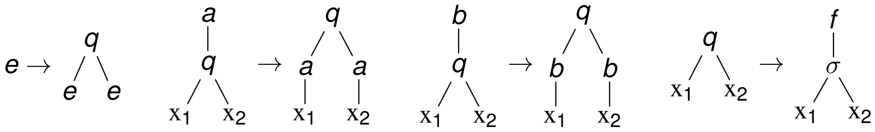


Example (Derivation)

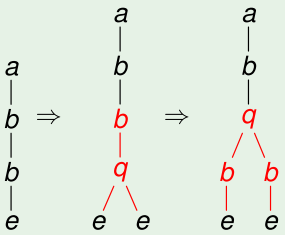


Example

States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules

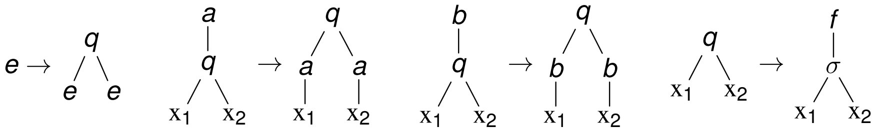


Example (Derivation)

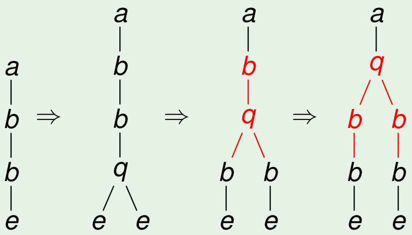


Example

States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules

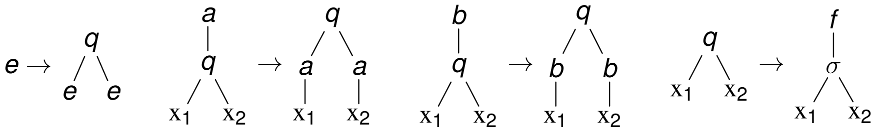


Example (Derivation)

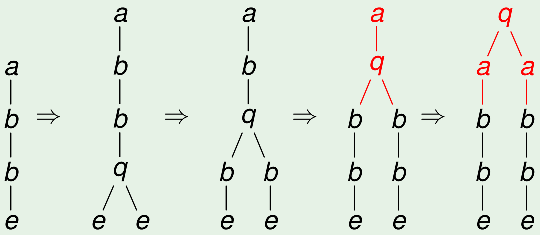


Example

States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules

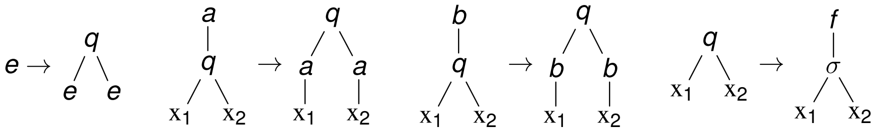


Example (Derivation)

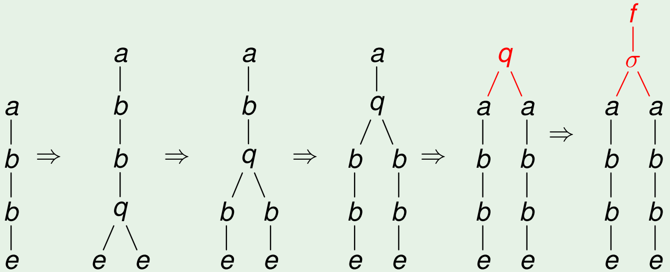


Example

States $\{f^{(1)}, q^{(2)}\}$ with final state f and rules



Example (Derivation)



Semantics

Definition

Computed transformation:

$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in F \colon t \Rightarrow^* q(u)\}$$

Semantics

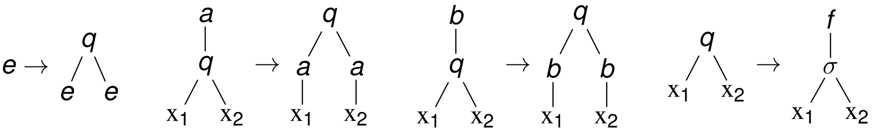
Definition

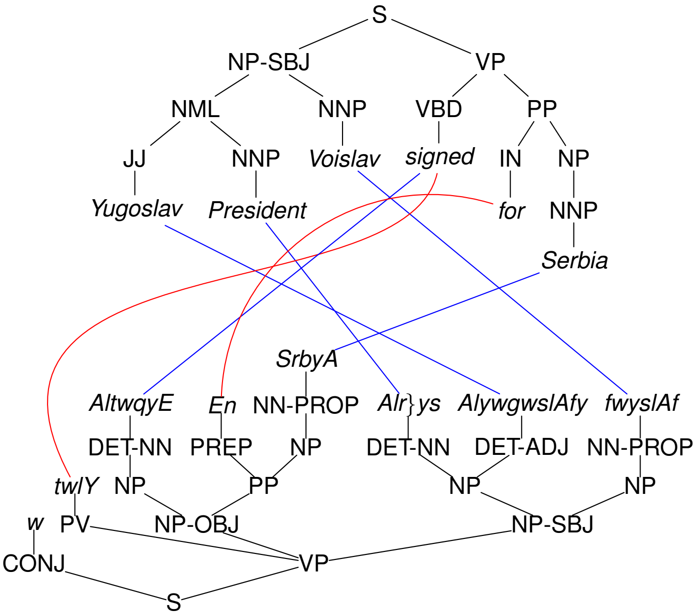
Computed transformation:

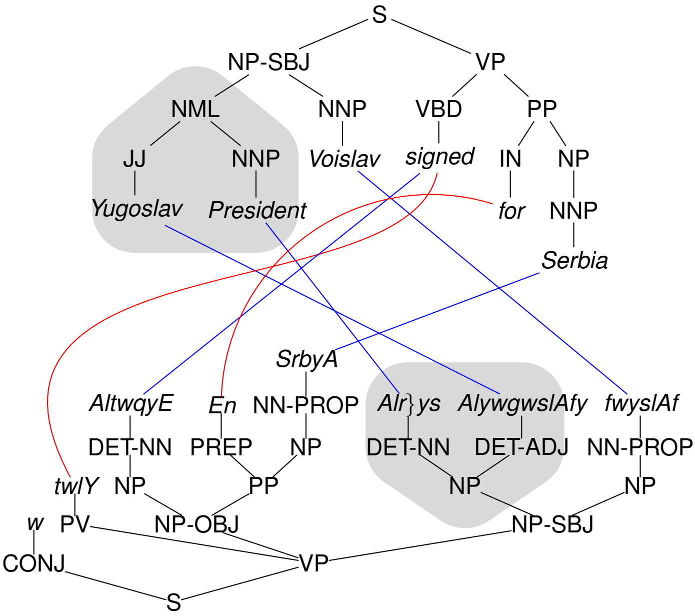
$$\tau_M = \{(t, u) \in T_\Sigma \times T_\Delta \mid \exists q \in F \colon t \Rightarrow^* q(u)\}$$

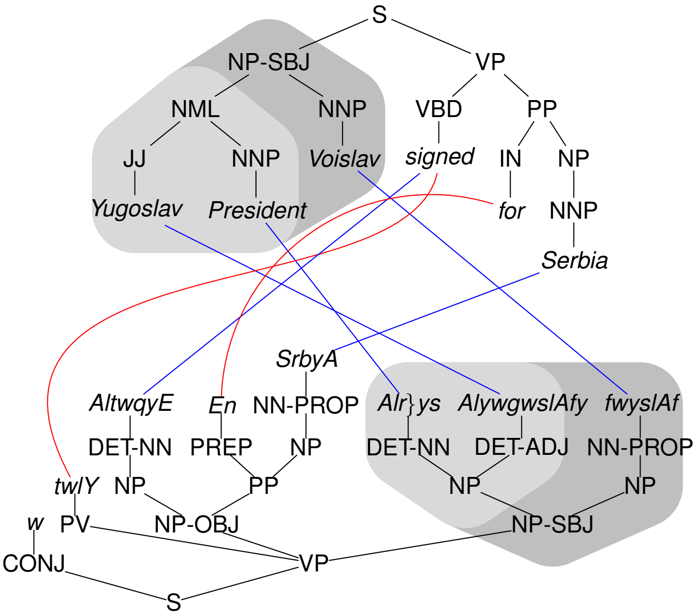
Example

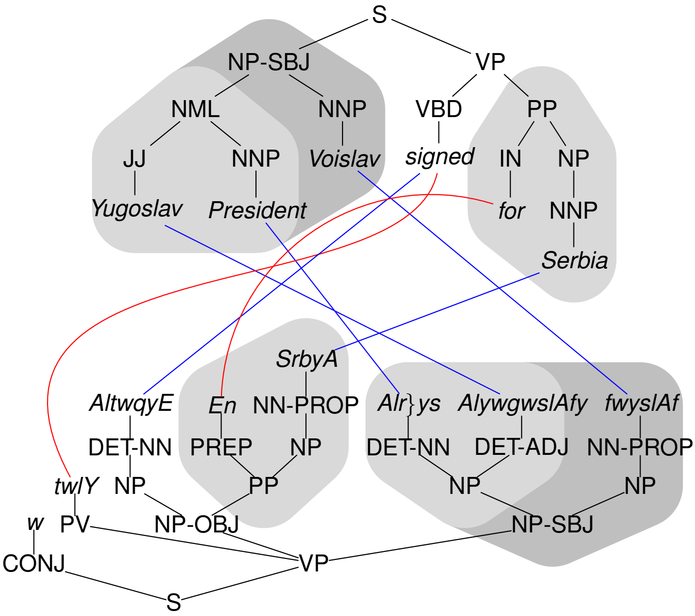
$$\tau_M = \{\langle t, \sigma(t, t)\rangle \mid t \in T_\Sigma\}$$

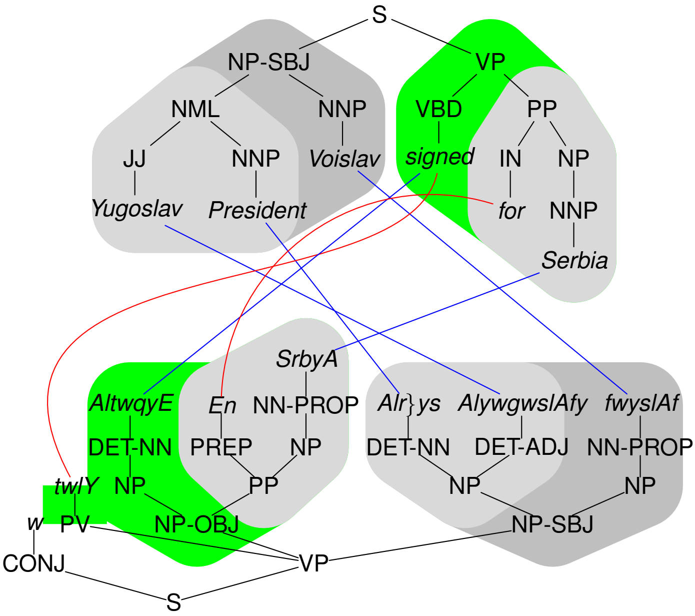



S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

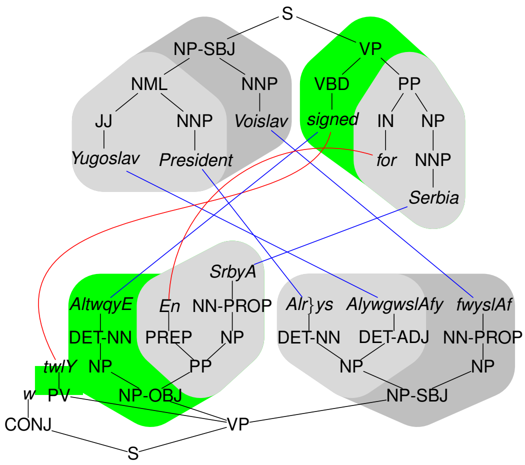

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

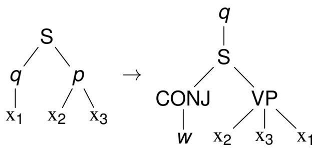

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S


S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

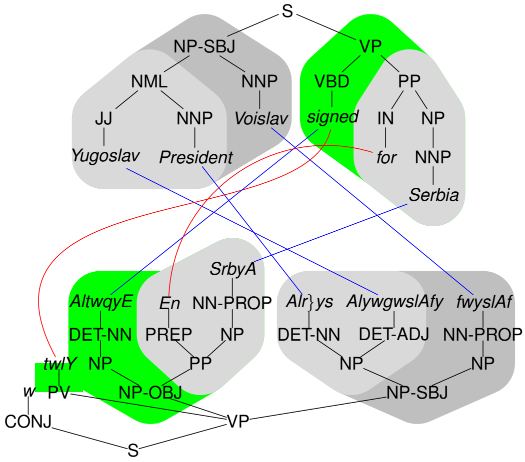

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

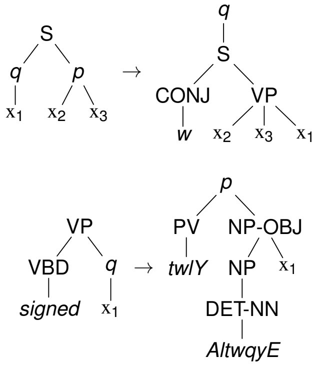
Rule extraction





Rule extraction





One-symbol normal form

Definition

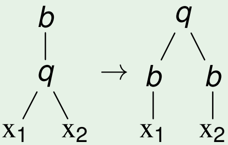
Rule in one-symbol normal form if it contains at most one symbol

One-symbol normal form

Definition

Rule in one-symbol normal form if it contains at most one symbol

Example (Engelfriet, Lilin, ∼ 2009)

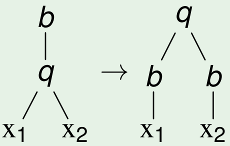


One-symbol normal form

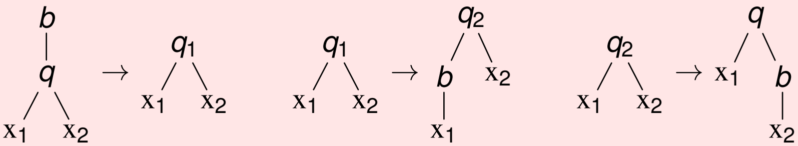
Definition

Rule in one-symbol normal form if it contains at most one symbol

Example (Engelfriet, Lilin, ~ 2009)



In one-symbol normal form



Basic properties

Example (Copying translation)

$$\tau_M = \{\langle t, \sigma(t, t)\rangle \mid t \in T_\Sigma\}$$

Consequences

- XMBOT are not symmetric
- XMBOT do not preserve regularity
- but they can be composed

Basic properties

Example (Copying translation)

$$\tau_M = \{\langle t, \sigma(t,t)\rangle \mid t \in T_\Sigma\}$$

Consequences

- XMBOT are not symmetric
- XMBOT do not preserve regularity
- but they can be composed

Basic properties

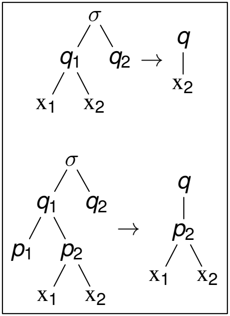
Example (Copying translation)

$$\tau_M = \{\langle t, \sigma(t, t)\rangle \mid t \in T_\Sigma\}$$

Consequences

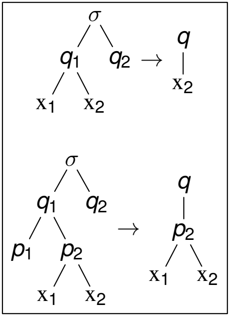
- XMBOT are not symmetric
- XMBOT do not preserve regularity
- but they can be composed

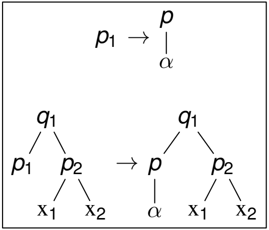
Composition



Simple composition works in the typical cases
[Baker 1979, Engelfriet 1975]

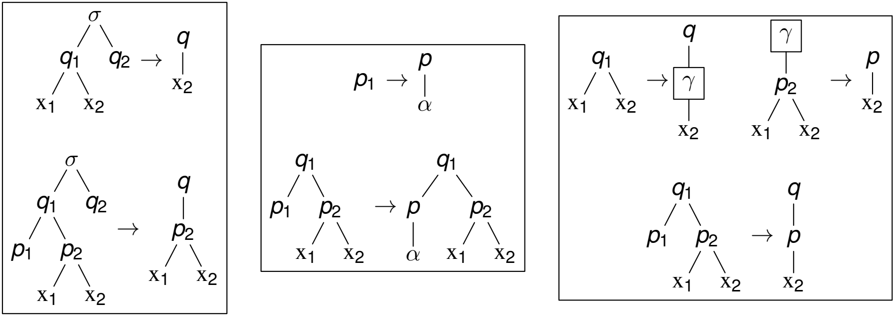
Composition





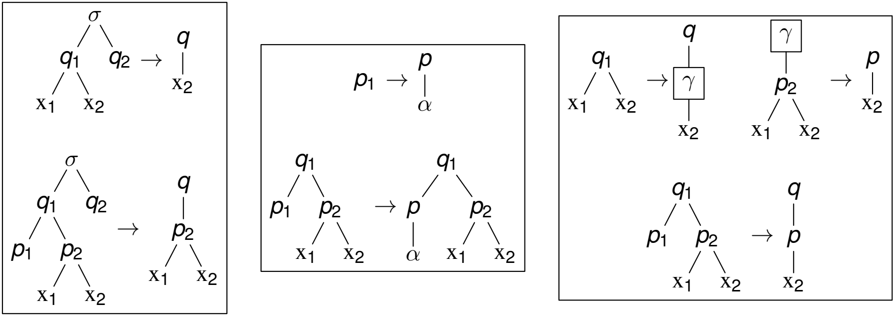
Simple composition works in the typical cases
[BAKER 1979, ENGELFRIET 1975]

Composition



Simple composition works in the typical cases
[BAKER 1979, ENGELFRIET 1975]

Composition



Simple composition works in the typical cases
[Baker 1979, Engelfriet 1975]

Summary

Model \ Criterion	EXPR	SYM	PRES	PRES^{-1}	COMP
Linear nondeleting TOP	✗	✗	✓	✓	✓
Linear nondeleting XTOP	✓	✓	✓	✓	✗
Linear nondeleting XMBOT	✓	✗	✗	✓	✓
Linear XMBOT	✓	✗	✗	✓	✓
General XMBOT	✓	✗	✗	✓	✗
reg.-preserving linear XMBOT	✓	✗	✓	✓	✓
invertable linear XMBOT	✓	✓	✓	✓	✓

Summary

Model \ Criterion	EXPR	SYM	PRES	PRES^{-1}	COMP
Linear nondeleting TOP	✗	✗	✓	✓	✓
Linear nondeleting XTOP	✓	✓	✓	✓	✗
Linear nondeleting XMBOT	✓	✗	✗	✓	✓
Linear XMBOT	✓	✗	✗	✓	✓
General XMBOT	✓	✗	✗	✓	✗
reg.-preserving linear XMBOT	✓	✗	✓	✓	✓
invertable linear XMBOT	✓	✓	✓	✓	✓

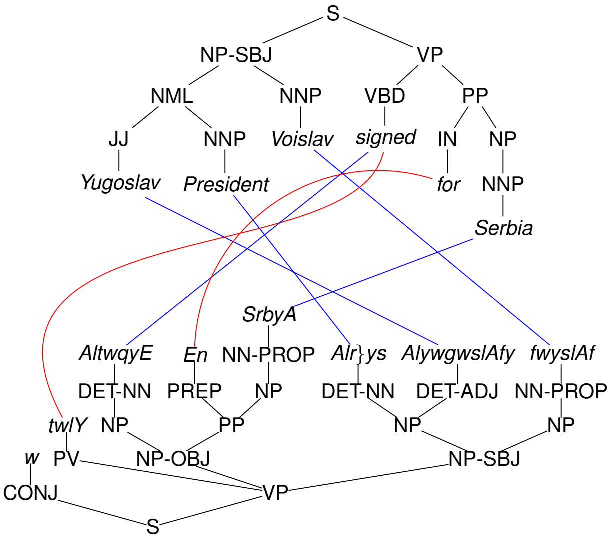
Implementation

No implementation yet,

Implementation

No implementation yet, but stay tuned

Synchronous Tree-Adjoining Grammars



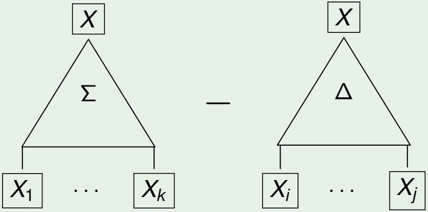
Syntax

Definition (Shieber, Schabes 1990)

Synchronous tree-adjoining grammar (STAG) is $G = (N, \Sigma, \Delta, S, R)$ with a finite set R of

- substitution rules
- adjunction rules

Example (Substitution rule)



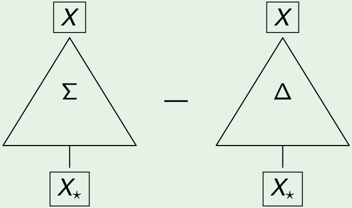
Syntax

Definition (Shieber, Schabes 1990)

Synchronous tree-adjoining grammar (STAG) is $G = (N, \Sigma, \Delta, S, R)$ with a finite set R of

- substitution rules
- adjunction rules

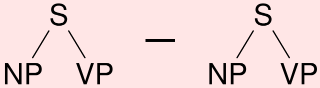
Example (Adjunction rule)



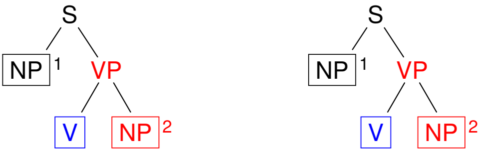
Example

Example

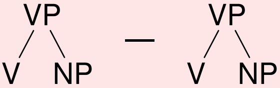
Used substitution rule



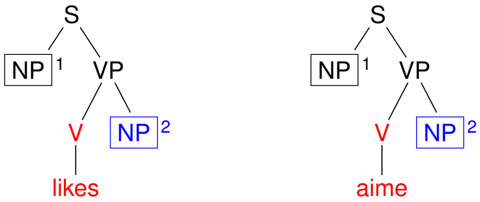
Example



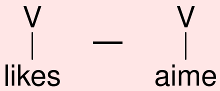
Used substitution rule



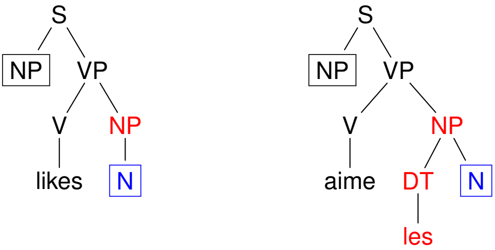
Example



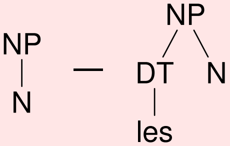
Used substitution rule



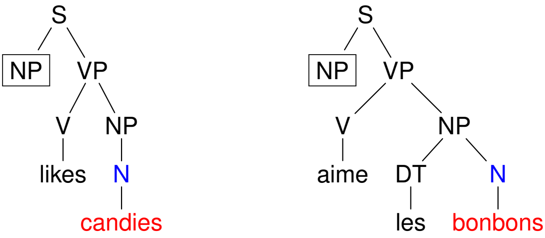
Example



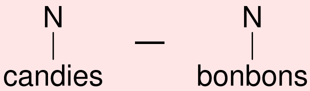
Used substitution rule



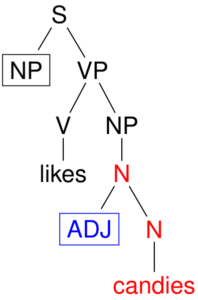
Example



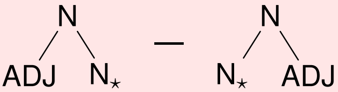
Used substitution rule



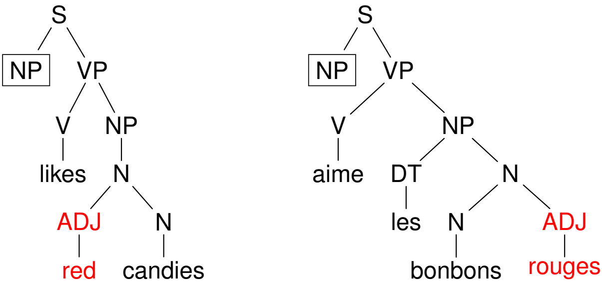
Example



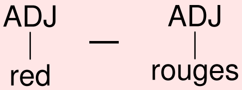
Used adjunction rule



Example



Used substitution rule



Semantics

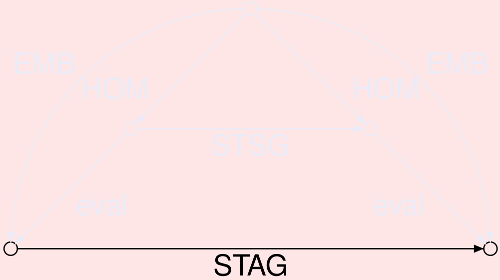
Definition

Computed transformation:

$$\tau_G = \{(t, u) \in T_\Sigma \times T_\Delta \mid (S, S) \Rightarrow^* (t, u)\}$$

Relation to tree transducers

Illustration



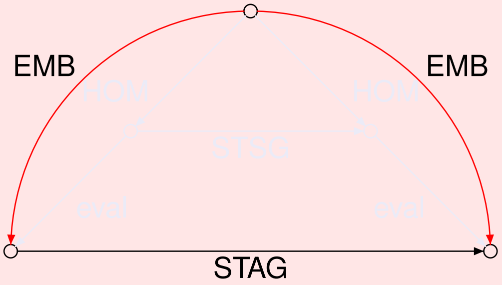
Definition (SHIEBER 2006)

embedded tree transducer is a macro tree transducer:

- linear, nondeleting, deterministic, total
- 1-parameter: linear, nondeleting

Relation to tree transducers

Illustration



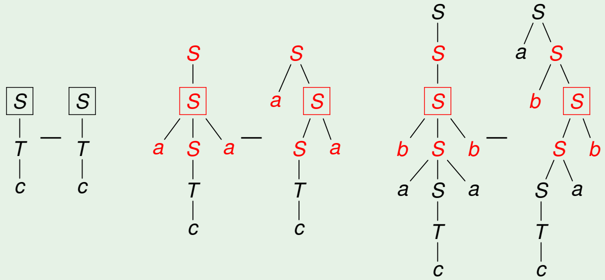
Definition (SHIEBER 2006)

embedded tree transducer is a macro tree transducer:

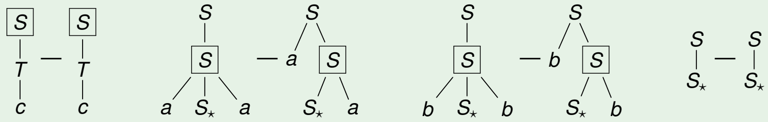
- linear, nondeleting, deterministic, total
- 1-parameter: linear, nondeleting

Copying example

Example

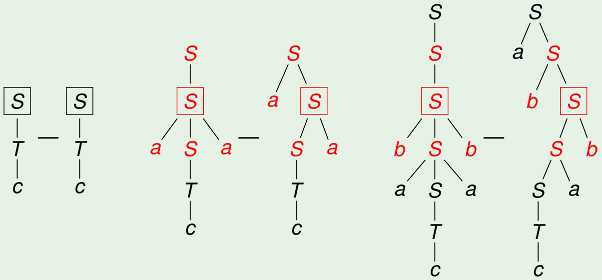


Example



Copying example

Example



String translation

$$\{(wcw^{\mathrm{R}}, wcw) \mid w \in \{a, b\}^*\}$$

Basic properties

Example (Copying translation)

$$\tau_G = \{(wcw^{\mathrm{R}}, wcw) \mid w \in \{a, b\}^*\}$$

Consequences

- STAG are symmetric
- STAG do not preserve regularity (neither direction)

Basic properties

Example (Copying translation)

$$\tau_G = \{(wcw^{\mathrm{R}}, wcw) \mid w \in \{a, b\}^*\}$$

Consequences

- STAG are symmetric
- STAG do not preserve regularity (neither direction)

Summary

Model \ Criterion	EXPR	SYM	PRES	PRES^{-1}	COMP
Linear nondeleting TOP	✗	✗	✓	✓	✓
Linear nondeleting XTOP	✓	✓	✓	✓	✗
Linear nondeleting XMBOT	✓	✗	✗	✓	✓
Linear XMBOT	✓	✗	✗	✓	✓
General XMBOT	✓	✗	✗	✓	✗
reg.-preserving linear XMBOT	✓	✗	✓	✓	✓
invertable linear XMBOT	✓	✓	✓	✓	✓
STAG	✓	✓	✗	✗	✗

Implementation

XTAG [The XTAG Project 2008]

- Implements TAG, STAG
- Optimized for natural language applications
- Application of STAG

http://www.cis.upenn.edu/~xtag/

Implementation

XTAG [The XTAG Project 2008]

- Implements TAG, STAG
- Optimized for natural language applications
- Application of STAG

`http://www.cis.upenn.edu/~xtag/`

References

- ARNOLD, DAUCHET: *Bi-transductions de forêts.* ICALP 1976
- BERSTEL, REUTENAUER: *Recognizable formal power series on trees.* Theor. Comput. Sci. 18, 1982
- ENGELFRIET: *Top-down tree transducers with regular look-ahead.* Math. Systems Theory 10, 1977
- GALLEY, HOPKINS, KNIGHT, MARCU: *What's in a translation rule?* HLT-NAACL 2004
- GRAEHL, KNIGHT: *Training tree transducers.* HLT-NAACL 2004
- MAY, KNIGHT: TIBURON — *a weighted tree automata toolkit.* CIAA 2006
- ROUNDS: *Mappings and grammars on trees.* Math. Systems Theory 4, 1970
- THATCHER *Generalized*^{2} *sequential machine maps.* J. Comput. System Sci. 4, 1970