How to train your multi bottom-up tree transducer

Andreas Maletti

Universität Stuttgart
Institute for Natural Language Processing
Stuttgart, Germany

andreas.maletti@ims.uni-stuttgart.de

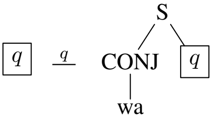
Portland, OR — June 22, 2011

Synchronous Tree Substitution Grammar

$\boxed{q}$ $\boxed{q}$

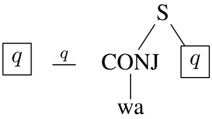
Used rule

Next rule

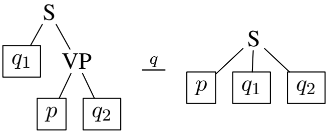


Synchronous Tree Substitution Grammar

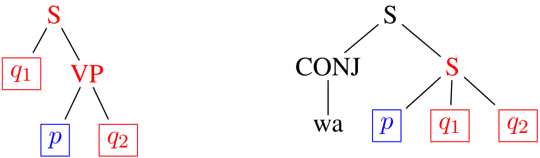
Used rule



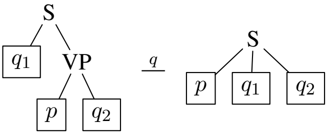
Next rule



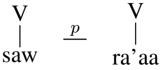
Synchronous Tree Substitution Grammar



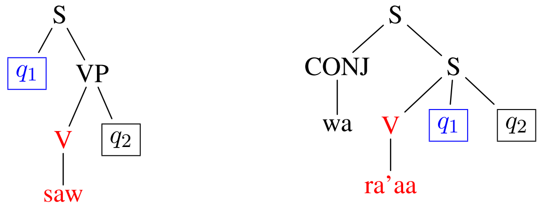
Used rule



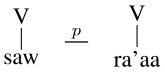
Next rule



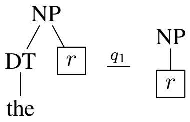
Synchronous Tree Substitution Grammar



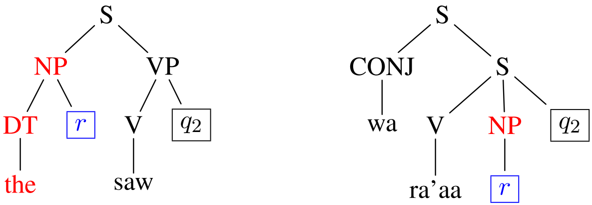
Used rule



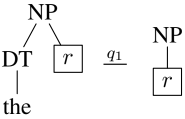
Next rule



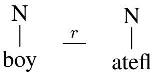
Synchronous Tree Substitution Grammar



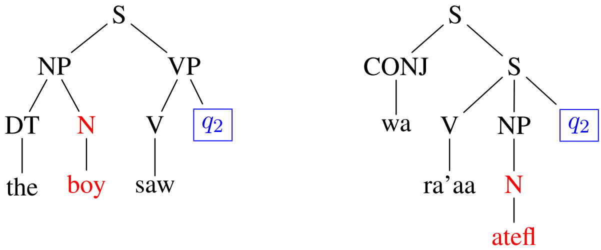
Used rule



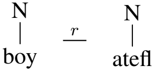
Next rule



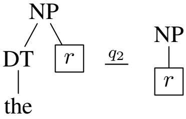
Synchronous Tree Substitution Grammar



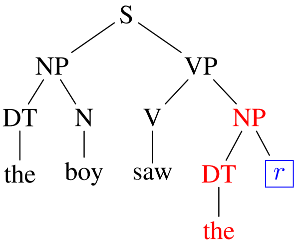
Used rule

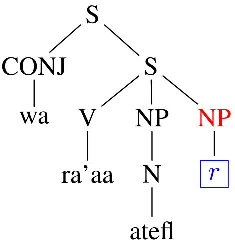


Next rule

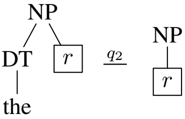


Synchronous Tree Substitution Grammar

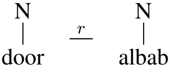




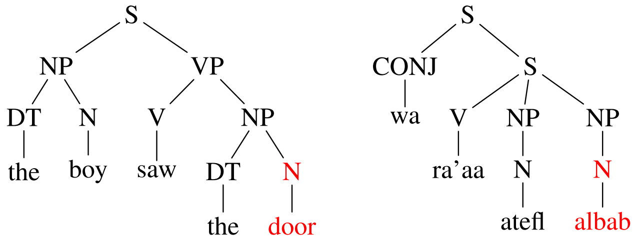
Used rule



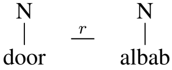
Next rule



Synchronous Tree Substitution Grammar



Used rule



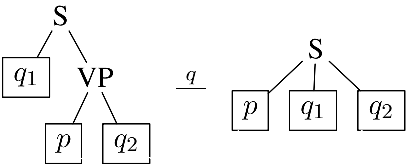
Next rule

Synchronous Tree Substitution Grammar

Rule shape

- states can occur only once in lhs and rhs
- states in rhs $=$ states in lhs
- exactly one lhs and one rhs

$\rightarrow$ bijective synchronization relation

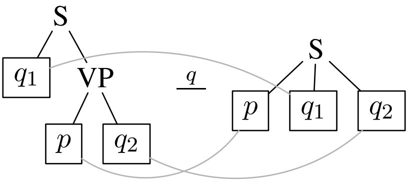


Synchronous Tree Substitution Grammar

Rule shape

- states can occur only once in lhs and rhs
- states in rhs = states in lhs
- exactly one lhs and one rhs

$\rightarrow$ bijective synchronization relation



Synchronous Tree Substitution Grammar

Rule shape

- states can occur only once in lhs and rhs
- states in rhs = states in lhs
- exactly one lhs and one rhs

$\rightarrow$ bijective synchronization relation

Notes

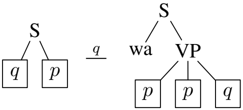
- version with states (instead of locality tests)
- equivalent to linear nondeleting extended tree transducers
- implemented in TIBURON [May, Knight 2006]

Multi Bottom-up Tree Transducer

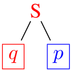
$\boxed{q}$ $\boxed{q}$

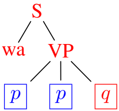
Used rule

Next rule

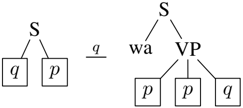


Multi Bottom-up Tree Transducer

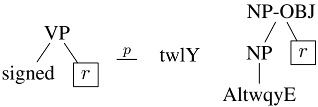




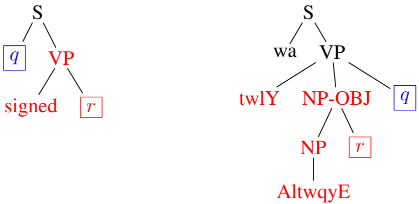
Used rule



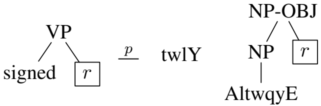
Next rule



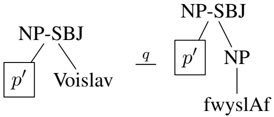
Multi Bottom-up Tree Transducer



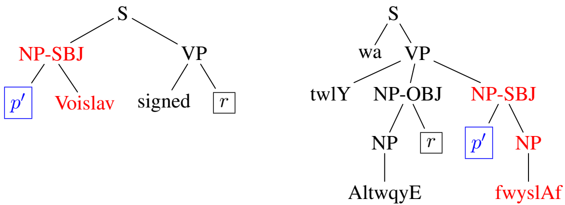
Used rule



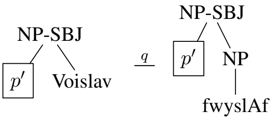
Next rule



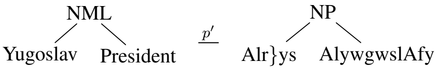
Multi Bottom-up Tree Transducer



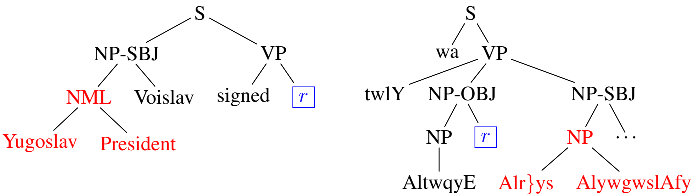
Used rule



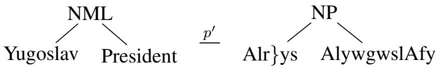
Next rule



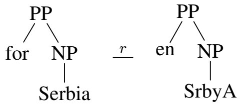
Multi Bottom-up Tree Transducer



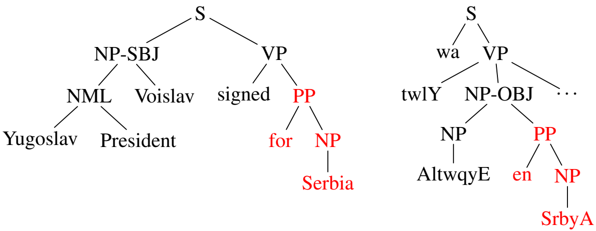
Used rule



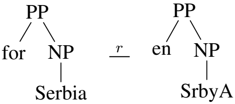
Next rule



Multi Bottom-up Tree Transducer



Used rule



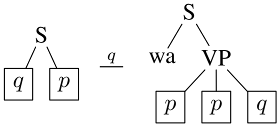
Next rule

Multi Bottom-up Tree Transducer

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
- STSG: states in rhs = states in lhs
  MBOT: states in rhs ⊆ states in lhs
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs

→ STSG: bijective synchronization relation
MBOT: inverse synchronization relation is functional

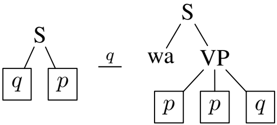


Multi Bottom-up Tree Transducer

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
- STSG: states in rhs $=$ states in lhs
  MBOT: states in rhs $\subseteq$ states in lhs
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs

$\rightarrow$ STSG: bijective synchronization relation
MBOT: inverse synchronization relation is functional

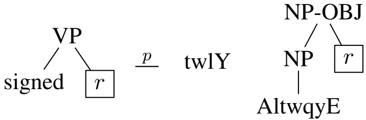


Multi Bottom-up Tree Transducer

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
- STSG: states in rhs $=$ states in lhs
  MBOT: states in rhs $\subseteq$ states in lhs
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs

$\rightarrow$ STSG: bijective synchronization relation
MBOT: inverse synchronization relation is functional

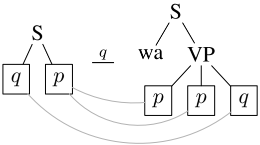


Multi Bottom-up Tree Transducer

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
- STSG: states in rhs = states in lhs
  MBOT: states in rhs ⊆ states in lhs
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs

→ STSG: bijective synchronization relation
MBOT: inverse synchronization relation is functional



STSG vs. MBOT

property	STSG	MBOT
simple and natural	✓	?
symmetric	✓	✗
preserves regularity	✓	✗
inverse pres. regularity	✓	✓
binarizable	✗	✓
closed under composition	✗	✓
input parsing	$\mathcal{O}(\lvert M\rvert n^x)$	$\mathcal{O}(\lvert M\rvert n^3)$
output parsing	$\mathcal{O}(\lvert M\rvert n^x)$	$\mathcal{O}(\lvert M\rvert n^y)$

where $x = 2\,\mathrm{rk}(M) + 5$ and
$y = 2\,\mathrm{rk}(M) + 2$

Additional Expressive Power

Finite copying

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

- ✓ desirable [ATANLP participants, 2010]
  e.g., for cross-serial dependencies
- ✗ harms preservation of regularity

Non-contiguous rules

- ✓ $+0.64$ BLEU [Sun, Zhang, Tan, 2009] $25.92 \rightarrow 26.56$
- ✓ do not harm preservation of regularity

Additional Expressive Power

Finite copying

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

- ✓ desirable [ATANLP participants, 2010]
  e.g., for cross-serial dependencies
- ✗ harms preservation of regularity

Non-contiguous rules

- ✓ $+0.64$ BLEU [Sun, Zhang, Tan, 2009] $25.92 \to 26.56$
- ✓ do not harm preservation of regularity

Approach

Old approach

1. extract STSG rules
2. train STSG
3. convert to MBOT
4. work with MBOT

New approach

1. extract (restricted) MBOT rules
2. train MBOT
3. work with MBOT

Approach

Old approach

1. extract STSG rules
2. train STSG
3. convert to MBOT
4. work with MBOT

New approach

1. extract (restricted) MBOT rules
2. train MBOT
3. work with MBOT

Roadmap

1. Motivation
2. Relation to STSSG
3. Rule Extraction and Training
4. Preservation of Recognizability

Synchronous Tree-Sequence Substitution Grammar

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
  STSSG: (no restriction)
- STSG: states in rhs = states in lhs
  MBOT: states in rhs ⊆ states in lhs
  STSSG: (no restriction)
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs
  STSSG: (no restriction)

Synchronous Tree-Sequence Substitution Grammar

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
  STSSG: (no restriction)
- STSG: states in rhs $=$ states in lhs
  MBOT: states in rhs $\subseteq$ states in lhs
  STSSG: (no restriction)
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs
  STSSG: (no restriction)

Synchronous Tree-Sequence Substitution Grammar

Rule shape

- STSG: states can occur only once in lhs and rhs
  MBOT: states can occur only once in lhs
  STSSG: (no restriction)
- STSG: states in rhs $=$ states in lhs
  MBOT: states in rhs $\subseteq$ states in lhs
  STSSG: (no restriction)
- STSG: exactly one lhs and one rhs
  MBOT: exactly one lhs
  STSSG: (no restriction)

Expressive Power

Corollary

$$\text{STSG} \subseteq \text{MBOT} \subseteq \text{STSSG}$$

Theorem

$$\text{STSG} \subset \text{MBOT} \subset \text{STSSG}$$

Claim

$$\text{MBOT}^{-1} ; \text{MBOT} = \text{STSSG}$$

Expressive Power

Corollary

$$\mathrm{STSG} \subseteq \mathrm{MBOT} \subseteq \mathrm{STSSG}$$

Theorem

$$\mathrm{STSG} \subset \mathrm{MBOT} \subset \mathrm{STSSG}$$

Claim

$$\mathrm{MBOT}^{-1} \,;\, \mathrm{MBOT} = \mathrm{STSSG}$$

Expressive Power

Corollary

$$\text{STSG} \subseteq \text{MBOT} \subseteq \text{STSSG}$$

Theorem

$$\text{STSG} \subset \text{MBOT} \subset \text{STSSG}$$

Claim

$$\text{MBOT}^{-1} \,;\, \text{MBOT} = \text{STSSG}$$

STSG vs. MBOT vs. STSSG

property	STSG	MBOT	STSSG
simple and natural	✓	?	??
symmetric	✓	✗	✓
preserves regularity	✓	✗	✗
inverse pres. regularity	✓	✓	✗
binarizable	✗	✓	✓
closed under composition	✗	✓	✗
input parsing	$\mathcal{O}(\lvert M\rvert n^x)$	$\mathcal{O}(\lvert M\rvert n^3)$	$\mathcal{O}(\lvert M\rvert n^y)$
output parsing	$\mathcal{O}(\lvert M\rvert n^x)$	$\mathcal{O}(\lvert M\rvert n^y)$	$\mathcal{O}(\lvert M\rvert n^y)$

where $x = 2\,\mathrm{rk}(M) + 5$ and
$y = 2\,\mathrm{rk}(M) + 2$

Roadmap

1. Motivation
2. Relation to STSSG
3. Rule Extraction and Training
4. Preservation of Recognizability

Input

- parallel bi-text
- parses for input and output
- word alignment

Output

- MBOT rules

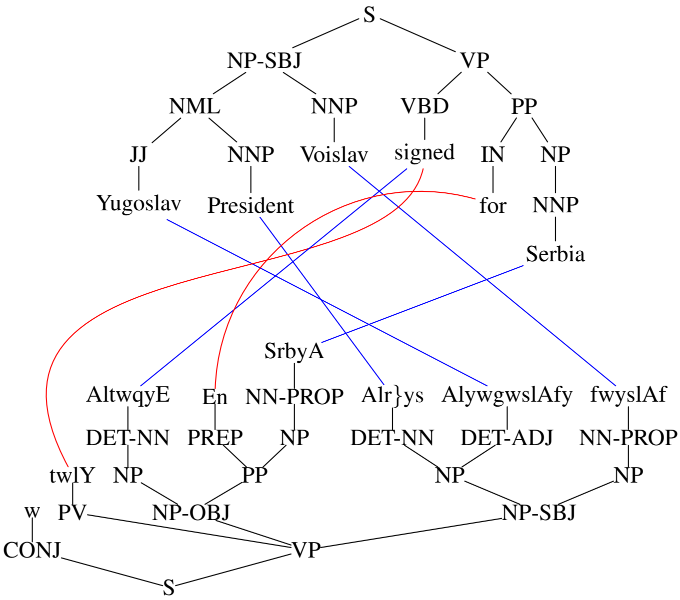
Rule Extraction

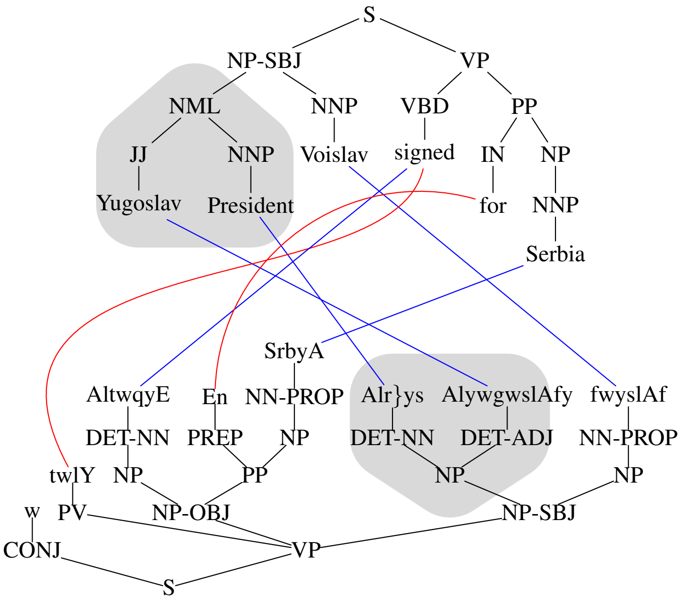
Input

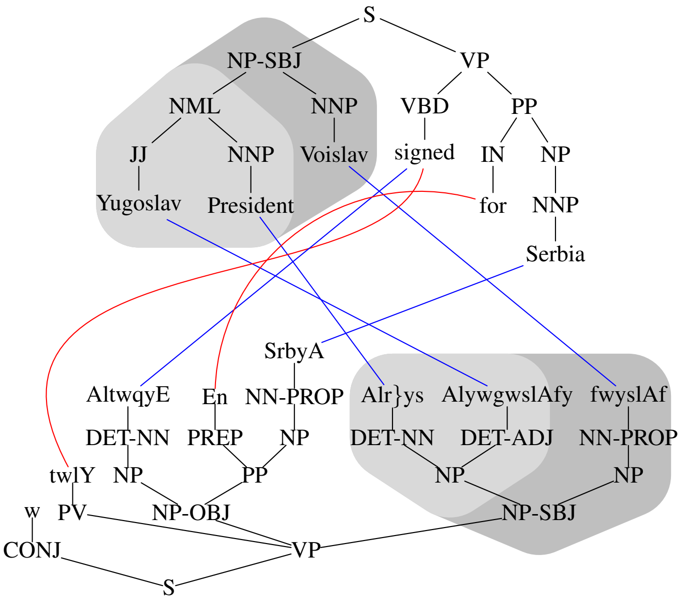
- parallel bi-text
- parses for input and output
- word alignment

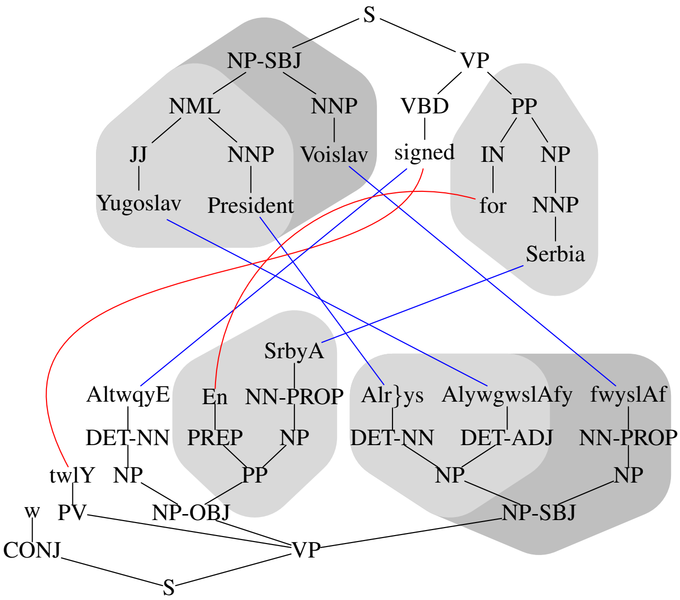
Output

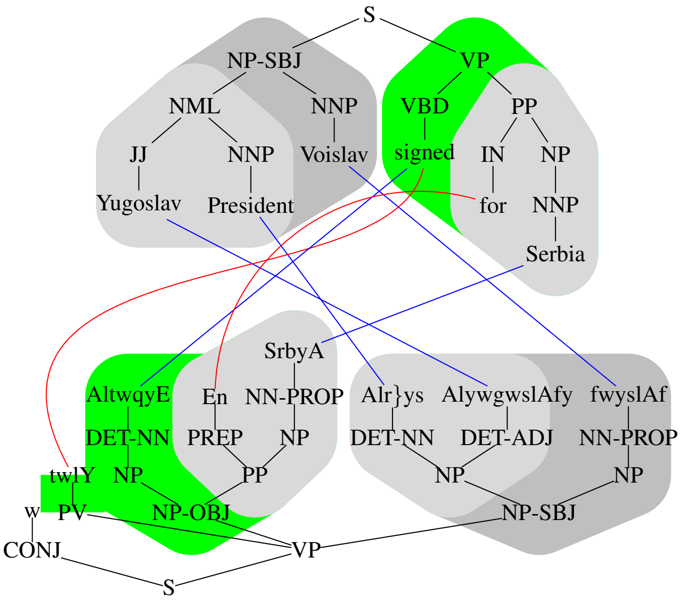
- MBOT rules


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

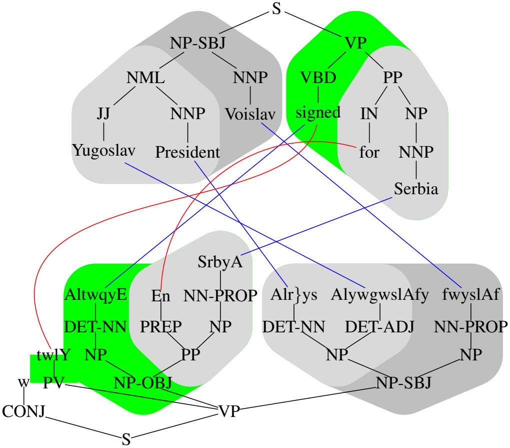

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

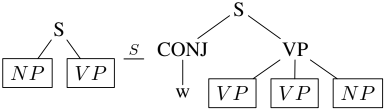

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S


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

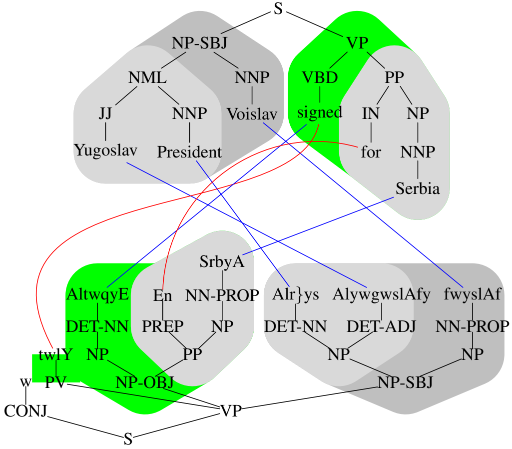

S
NP-SBJ
VP
NML
NNP
VBD
PP
JJ
NNP
Voislav
signed
IN
NP
Yugoslav
President
for
NNP
Serbia
SrbyA
AltwqyE
En
NN-PROP
Alr}ys
AlywgwslAfy
fwyslAf
DET-NN
PREP
NP
DET-NN
DET-ADJ
NN-PROP
twlY
NP
PP
NP
NP
w
PV
NP-OBJ
NP-SBJ
CONJ
VP
S

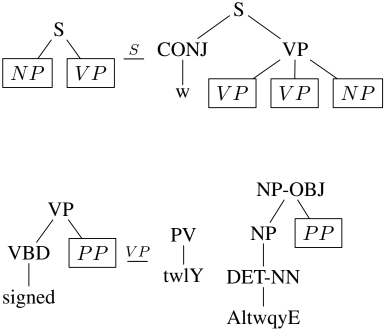
Rule Extraction





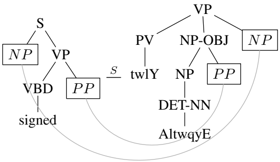
Rule Extraction





Extracted Rules

STSG rule

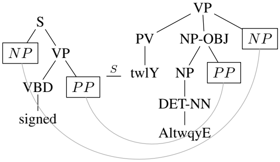


MBOT rules

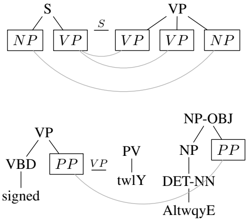
STSSG rule

Extracted Rules

STSG rule



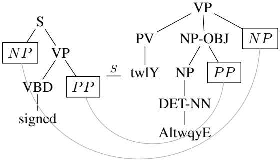
MBOT rules



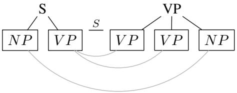
STSSG rule

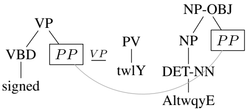
Extracted Rules

STSG rule

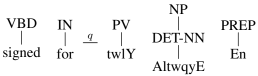


MBOT rules





STSSG rule



Definition

A tree language is regular if it can be represented by a TSG (with states).

Example

$$\{\sigma(t, t) \mid t \in T_\Sigma\}$$

is not regular.

Theorem

The set of derivations of an MBOT is regular.

Conclusion

Minimal adaptation of [Graehl, Knight, May 2008] can be used.

Theorem

The set of derivations of an MBOT is regular.

Conclusion

Minimal adaptation of [Graehl, Knight, May 2008] can be used.

Roadmap

Preservation of Regularity

Input

- MBOT M
- regular tree language L

Question

Is $\quad M(L) = \{t \mid \exists s \in L\colon \langle s, t\rangle \in M\} \quad$ regular?

Example

- $M = \{\langle t, \sigma(t, t)\rangle \mid t \in T_\Sigma\}$
- infinite, but regular tree language $L \subseteq T_\Sigma$

Then $M(L) = \{\sigma(t, t) \mid t \in L\}$ is not regular.

Preservation of Regularity

Input

- MBOT M
- regular tree language L

Question

Is $M(L) = \{t \mid \exists s \in L \colon \langle s, t \rangle \in M\}$ regular?

Example

- $M = \{\langle t, \sigma(t, t) \rangle \mid t \in T_\Sigma\}$
- infinite, but regular tree language $L \subseteq T_\Sigma$

Then $M(L) = \{\sigma(t, t) \mid t \in L\}$ is not regular.

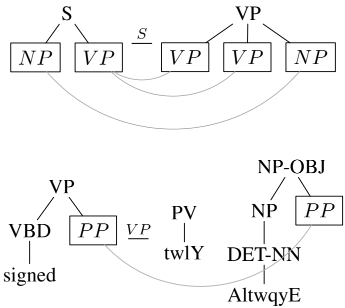
Preservation of Regularity

Why desired?

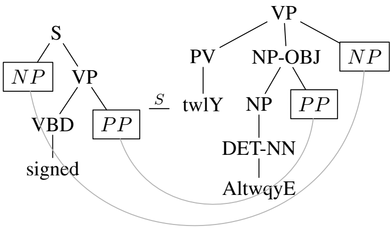
- easy and efficient representation of output languages
- allows intersection with (regular) language model
- allows bucket-brigade algorithms
  [May, Knight, Vogler, 2010]

Rule Composition

Input rules



Output rule



Rule Composition

Power MBOT

$$M^k = \{R_1 \circ \cdots \circ R_k \mid R_1, \ldots, R_k \in M\}$$

Explanation

- M^k contains k-fold composed rules
- composition fails if states do not match
- composition succeeds (with no effect) if no matchable states present

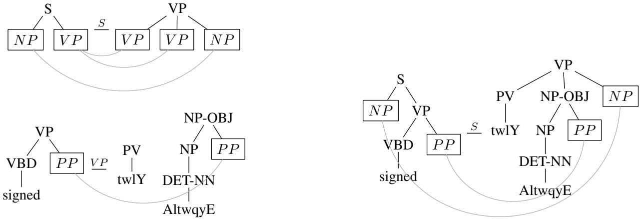
Finitely Collapsing

Definition

M is finitely collapsing if there exists $k \in \mathbb{N}$ such that

- every state occurs at most once in rhs of every rule of M^k.

Example



Finite Synchronization

Definition

M has finite synchronization if there exists $k \in \mathbb{N}$ such that

- all occurrences of a state occur in the same tree in the rhs of every rule of M^k.

Theorem (Raoult, 1997)

MBOT with finite synchronization and finitely collapsing MBOT preserve regularity.

Finite Synchronization

Definition

M has finite synchronization if there exists $k \in \mathbb{N}$ such that

- all occurrences of a state occur in the same tree in the rhs of every rule of M^k.

Theorem (Raoult, 1997)

MBOT with finite synchronization and finitely collapsing MBOT preserve regularity.

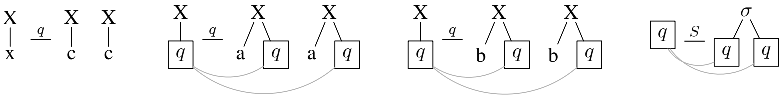
Copy-free

Definition

M is copy-free if there exists $k \in \mathbb{N}$ such that for every rule of M^k and all occurrences w of a state:

- w is the root of a tree in the rhs or
- w belongs to a fixed tree in the rhs.

Example (not copy-free)

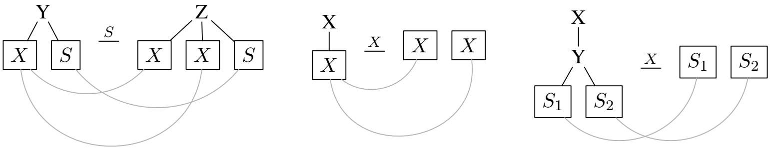


Definition

M is copy-free if there exists $k \in \mathbb{N}$ such that for every rule of M^k and all occurrences *w* of a state:

- *w* is the root of a tree in the rhs or
- *w* belongs to a fixed tree in the rhs.

Example (copy-free)



Main Result

Theorem

Copy-free MBOT preserve regularity.

Theorem

finitely collapsing $\subset$ *finite synchronization* $\subset$ *copy-free*

Note

All 3 restrictions can be guaranteed during rule extraction.

Main Result

Theorem

Copy-free MBOT preserve regularity.

Theorem

finitely collapsing $\subset$ *finite synchronization* $\subset$ *copy-free*

Note

All 3 restrictions can be guaranteed during rule extraction.

Main Result

Theorem
Copy-free MBOT preserve regularity.

Theorem
finitely collapsing $\subset$ *finite synchronization* $\subset$ *copy-free*

Note
All 3 restrictions can be guaranteed during rule extraction.

Thank you for your attention!

References

1. ARNOLD, DAUCHET: Morphismes et bimorphismes d'arbres.
   Theoret. Comput. Sci., 1982
2. ATANLP participants: Desired properties of syntax-based MT systems.
   ACL workshop, 2010
3. GRAEHL, KNIGHT, MAY: Training tree transducers.
   Computational Linguistics, 2008
4. MAY, KNIGHT: TIBURON — A weighted tree automata toolkit.
   In *CIAA*, 2006
5. MAY, KNIGHT, VOGLER: Efficient Inference through Cascades of Weighted Tree Transducers.
   In *ACL*, 2010
6. RAOULT: Rational tree relations.
   Bull. Belg. Math. Soc. Simon Stevin, 1997
7. SUN, ZHANG, TAN: A non-contiguous tree sequence alignment-based model for statistical machine translation.
   In *ACL*, 2009