Graph Transduction for Semantics

David Chiang & Kevin Knight
USC/ISI

MURI Review
November 22, 2013

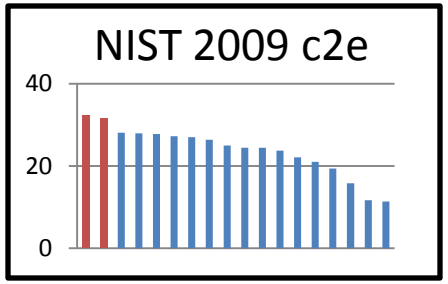
Machine Translation

Phrase-based MT

source string → target string

Syntax-based MT

source string → source tree → target tree → target string



Meaning-based MT

source string → source tree → meaning representation → target tree → target string

Why Meaning-Based MT?

- That's what translation is:
  - build grammatical target text…
  - that preserves the meaning of the source text

Oh, it got the meaning wrong…

- or -

It got the right meaning, but rendered it dis-fluently…

Isn't Meaning-Based MT Impossible?

- What content goes into the meaning representation?

  STAGES Abstract Meaning Representation

  & annotation MR/DEFT

- How are meaning representations probabilistically generated, transformed, scored, ranked?

  MURI automata theory, efficient algorithms

- How can a full MT system be built?

  BOLT engineering, modeling, features, training

Automata Frameworks

- How to represent and manipulate linguistic representations?
- Linguistics, NLP, and Automata Theory used to be together (1960s, 70s)
  - Context-free grammars were invented to model human language
  - Tree transducers were invented to model transformational grammar
- Renewed connections around MT (this century)
- Role: greatly simplify systems!

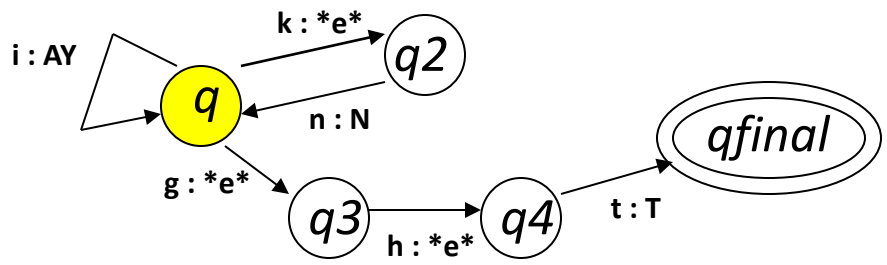
Finite-State Transducer (FST)

Original input:

k n i g h t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



Transformation:

q k n i g h t

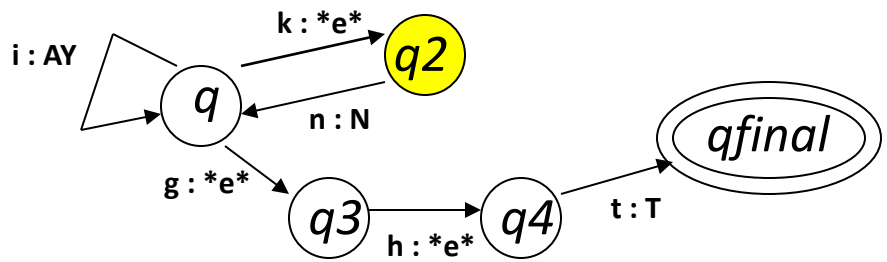
Finite-State (String) Transducer

Original input:

k n i g h t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



Transformation:

q2 n i g h t

Finite-State (String) Transducer

Original input:

k n i g h t

FST

q k → *q2* *e*

q i → *q* AY

q g → *q3* *e*

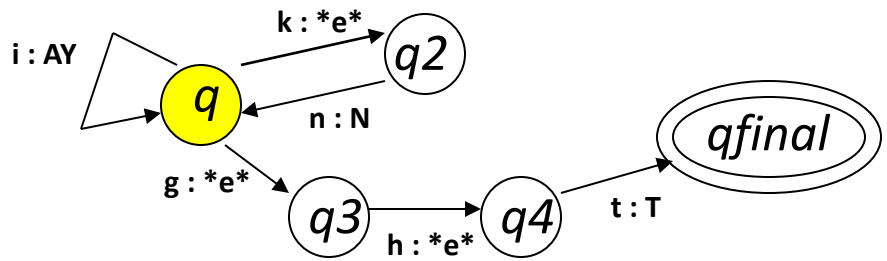
q2 n → *q* N

q3 h → *q4* *e*

q4 t → *qfinal* T

Transformation:

N *q* i g h t



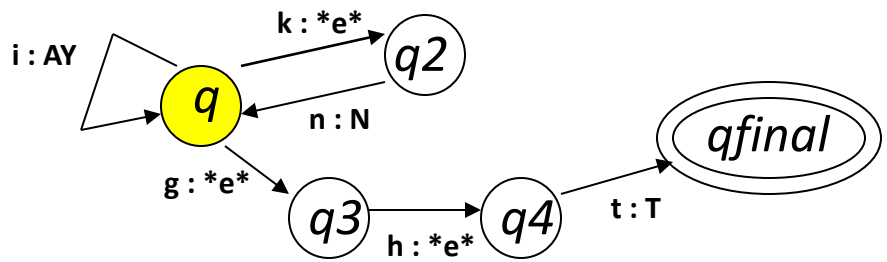
Finite-State (String) Transducer

Original input:

k n i g h t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



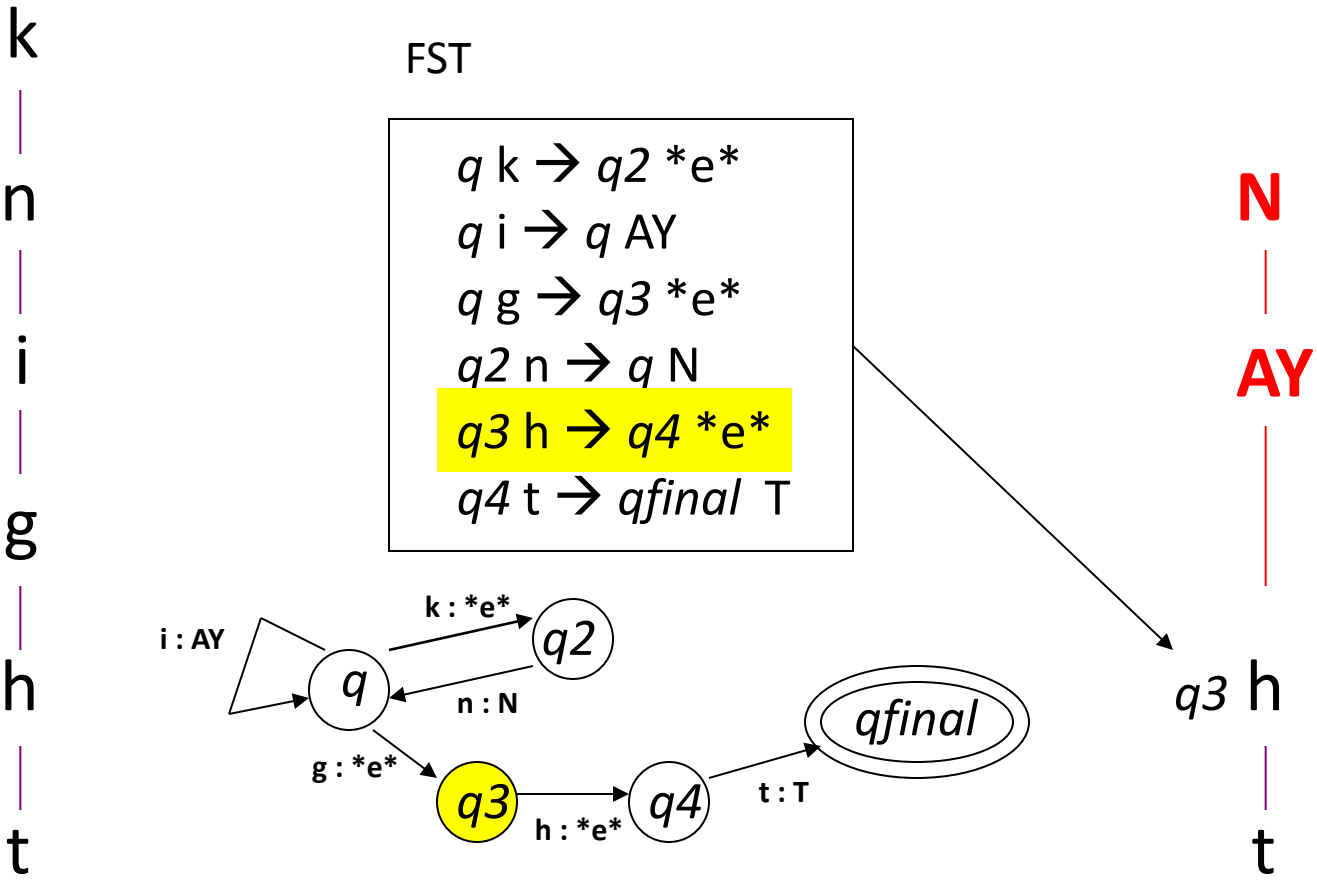
Transformation:

N AY *q* g h t

Finite-State (String) Transducer

Original input:

k n i g h t



Transformation:

N AY q3 h t

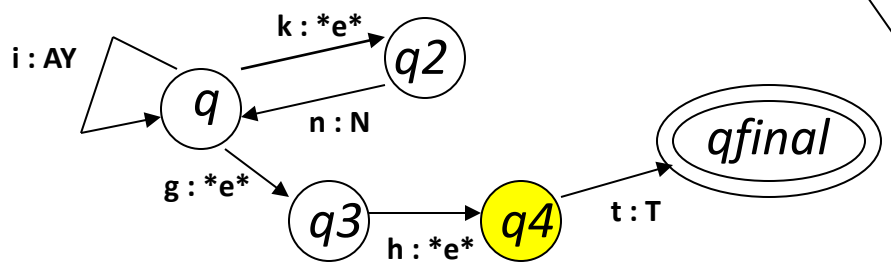
Finite-State (String) Transducer

Original input:

k
n
i
g
h
t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



Transformation:

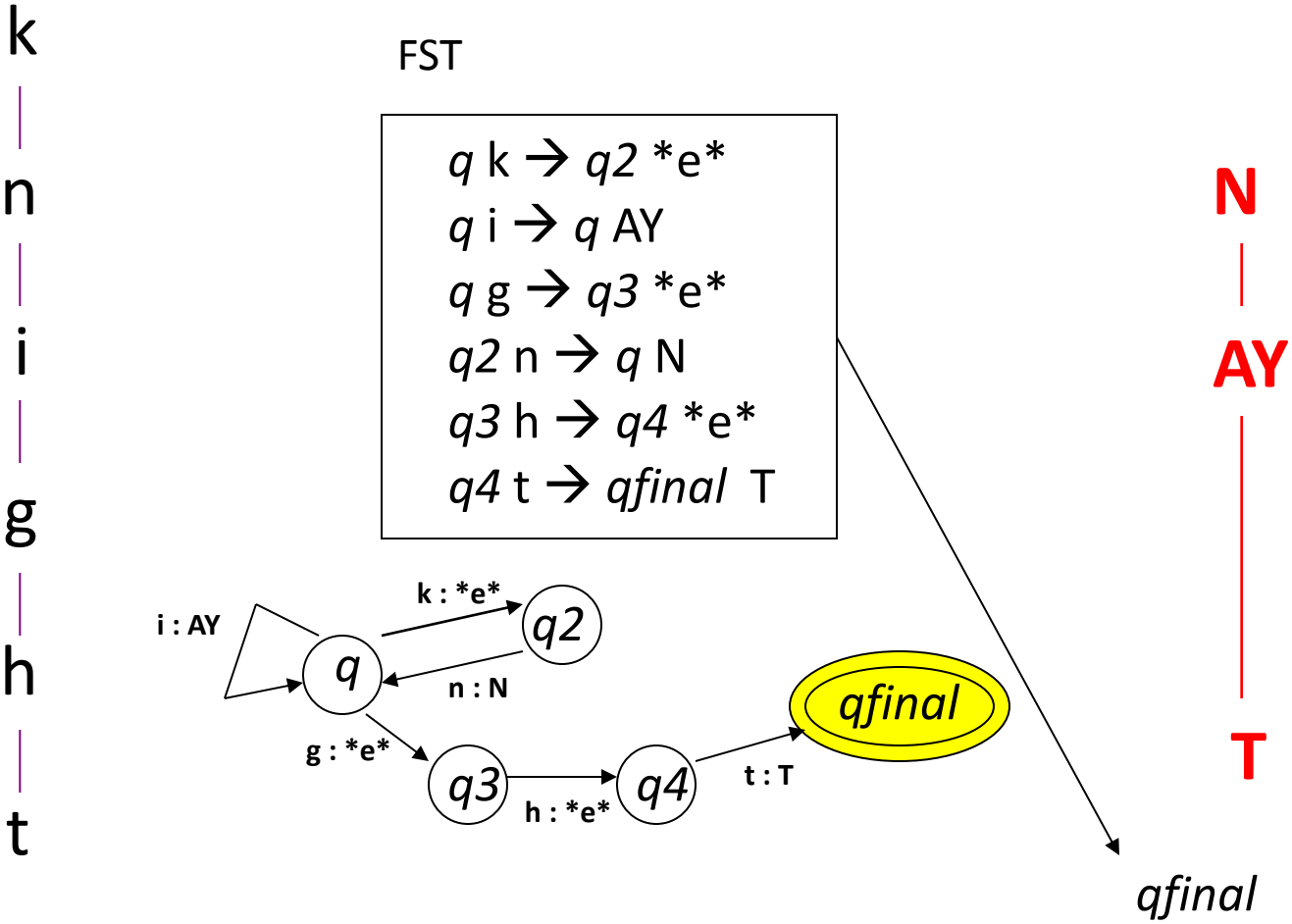
N
AY

q4 t

Finite-State (String) Transducer

Original input:

k
n
i
g
h
t



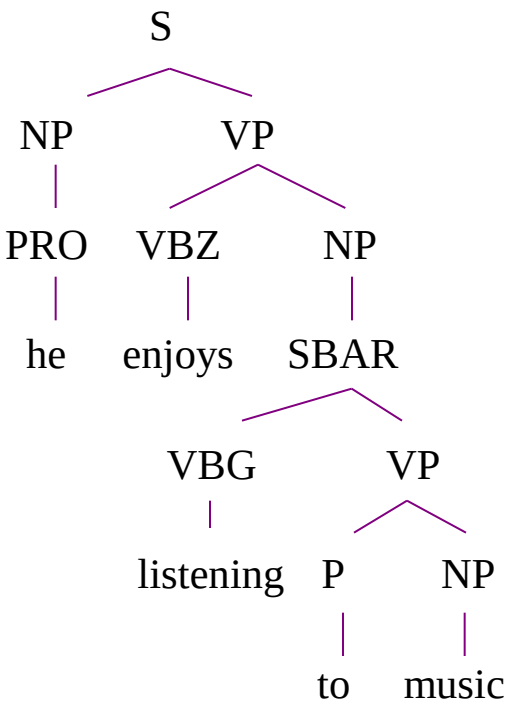
General-Purpose Algorithms for String Automata

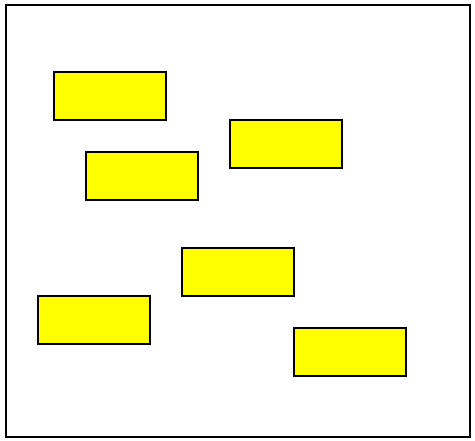
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)
EM training	Forward-backward EM (Baum & Welch, 1971; Eisner 2001)
Determinization …	… of weighted string acceptors (Mohri, 1997)
Intersection	WFSA intersection
Application	string → WFST → WFSA
Transducer composition	WFST composition (Pereira & Riley, 1996)
General-purpose toolkit	Carmel (Graehl & Knight 97), OpenFST (Google, via AT&T), ...

Top-Down Tree Transducer

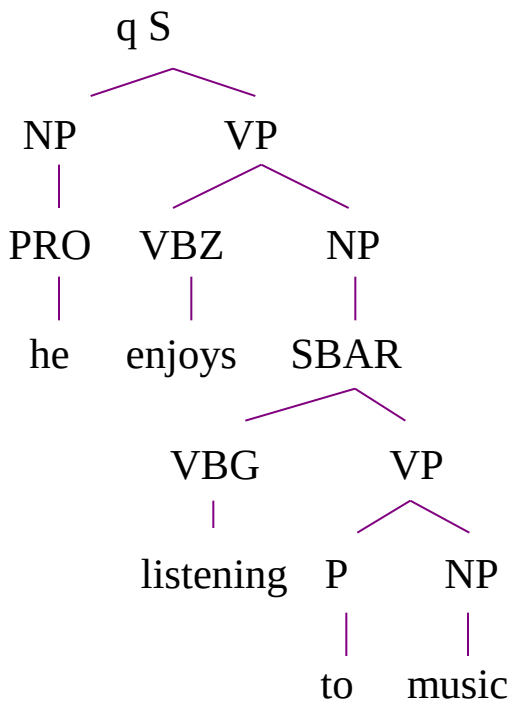
(W. Rounds 1970; J. Thatcher 1970)

Original input:





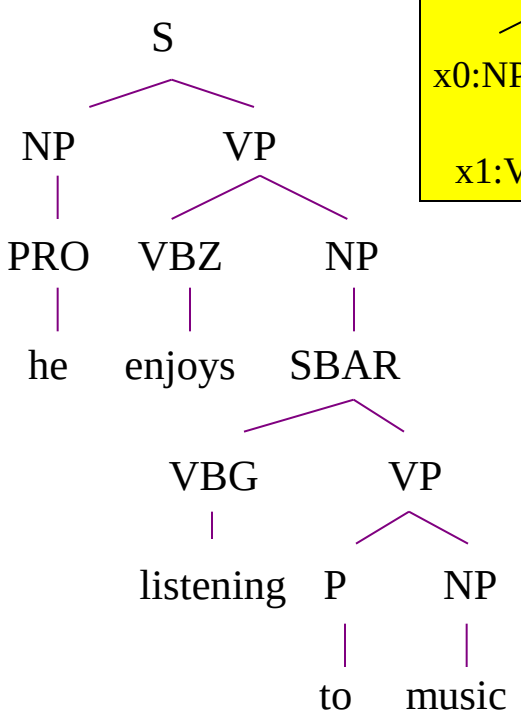
Transformation:

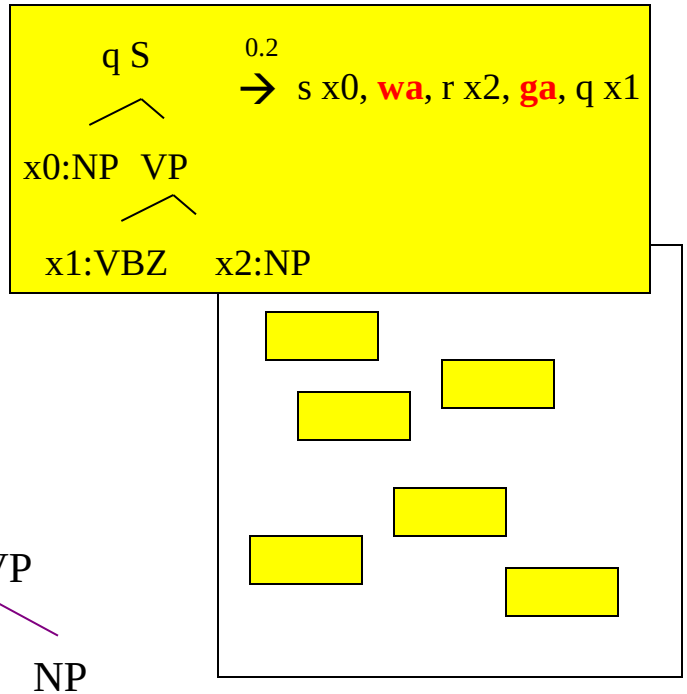


Top-Down Tree Transducer

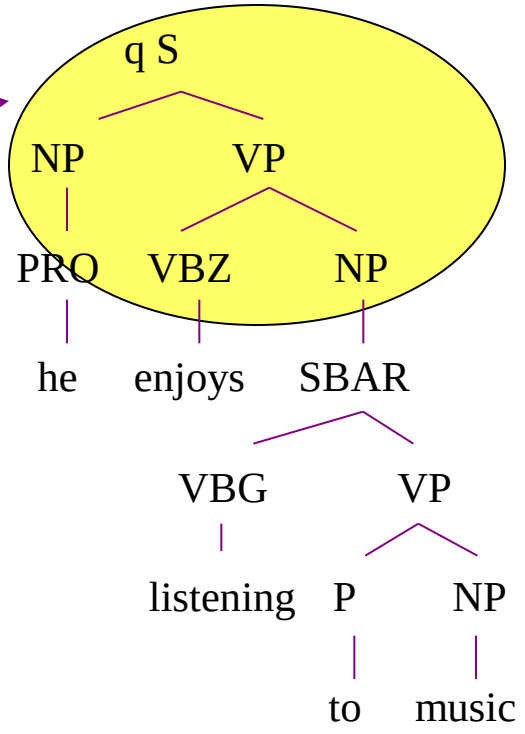
(W. Rounds 1970; J. Thatcher 1970)

Original input:





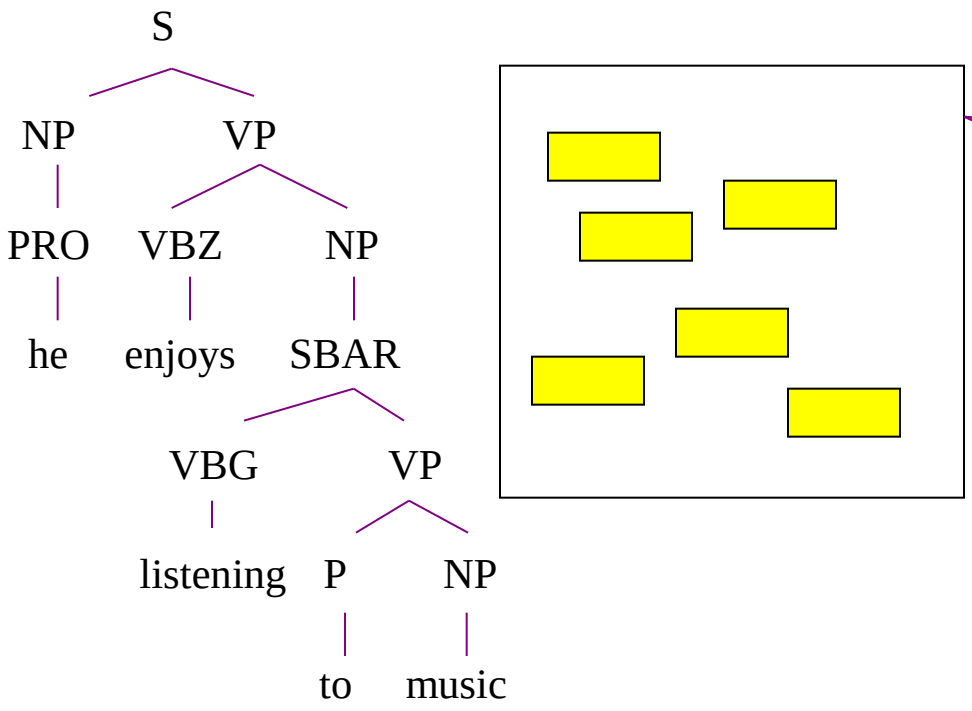
Transformation:



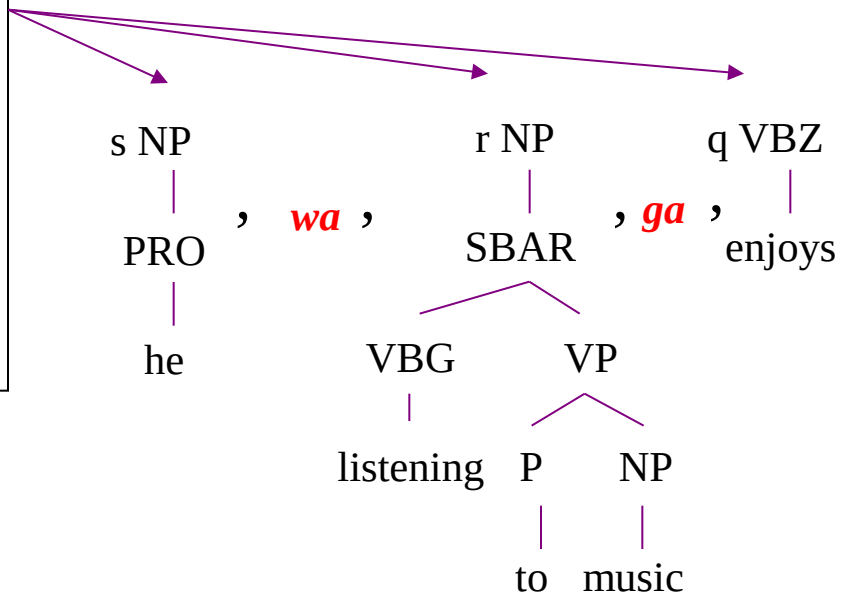
Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

Original input:

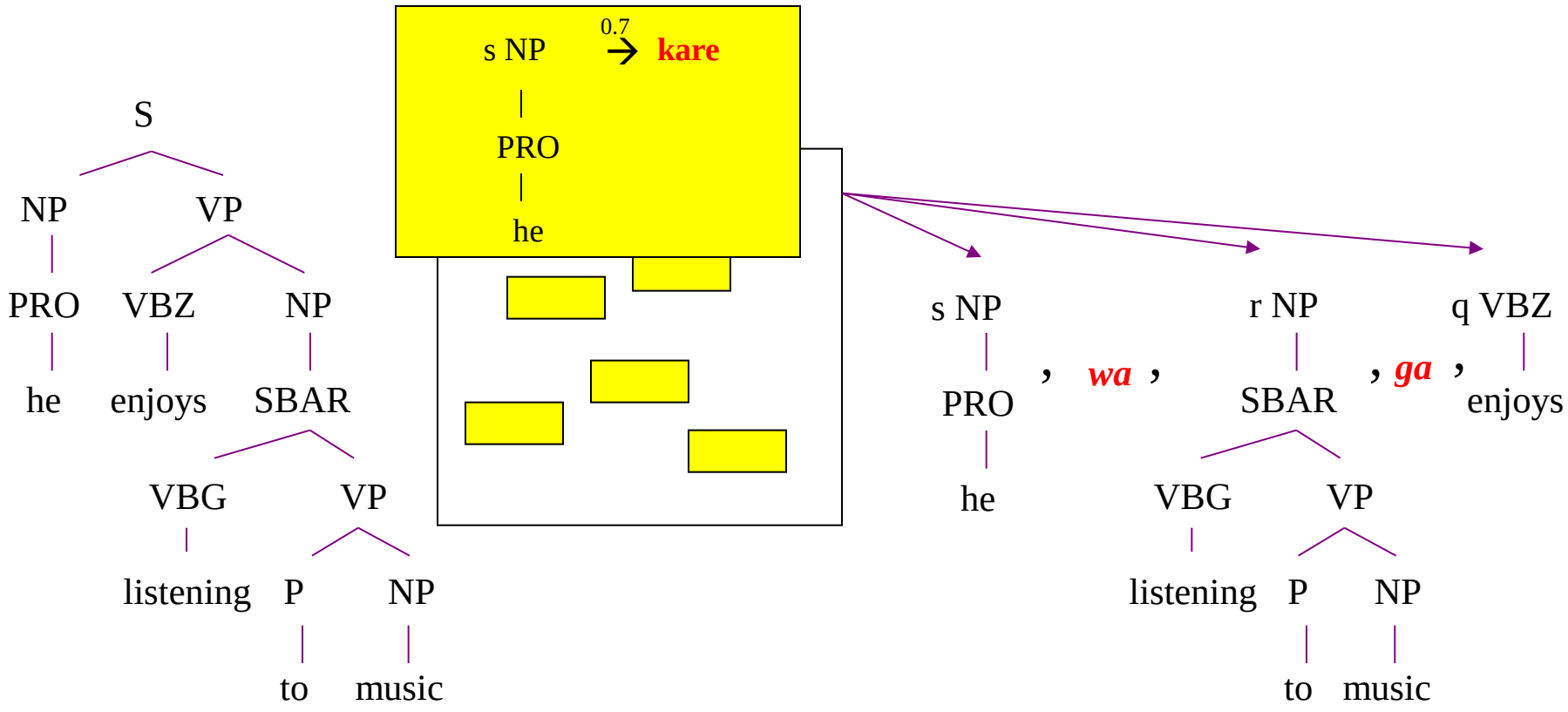


Transformation:



Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

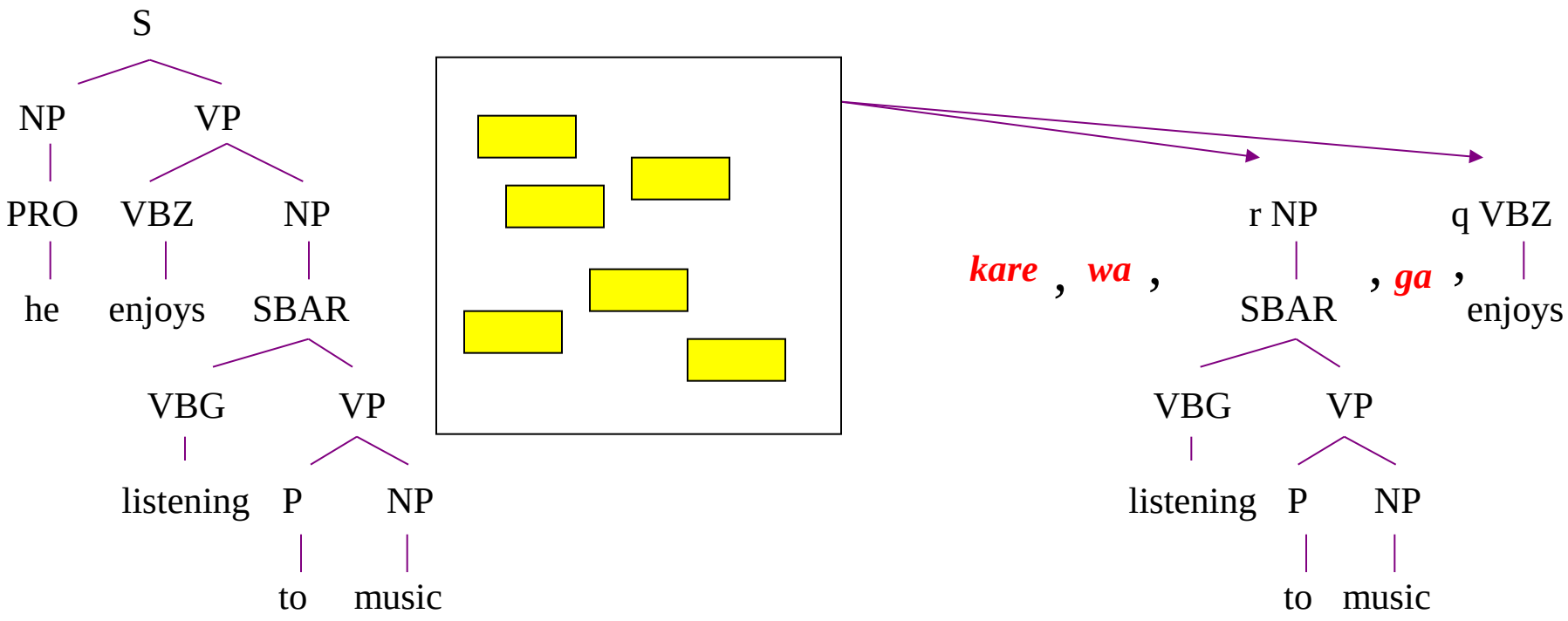


Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

Original input:

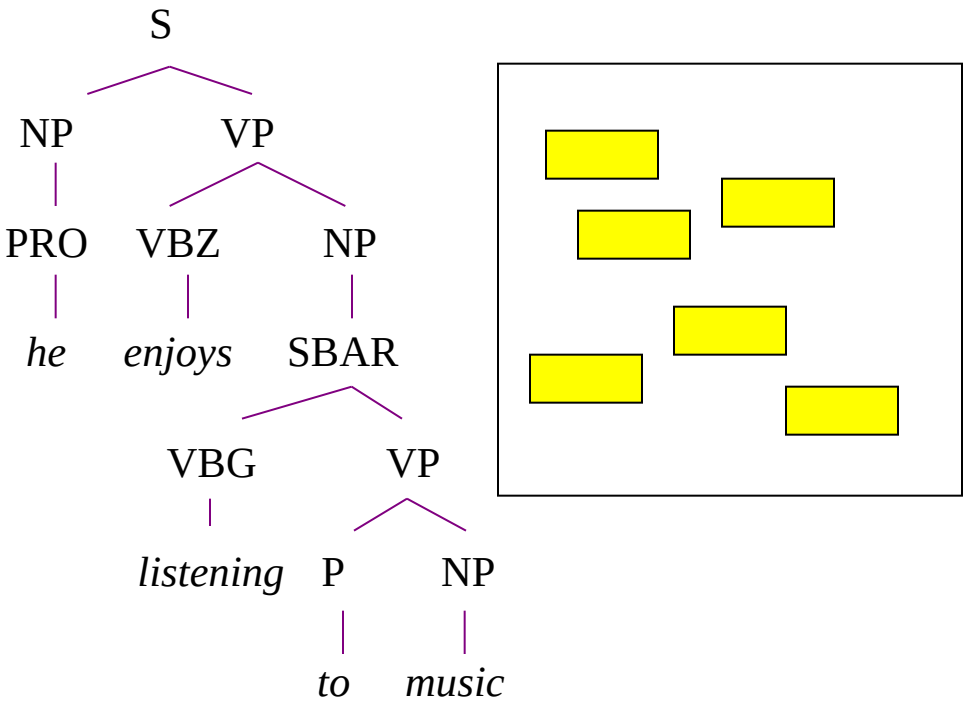
Transformation:



Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

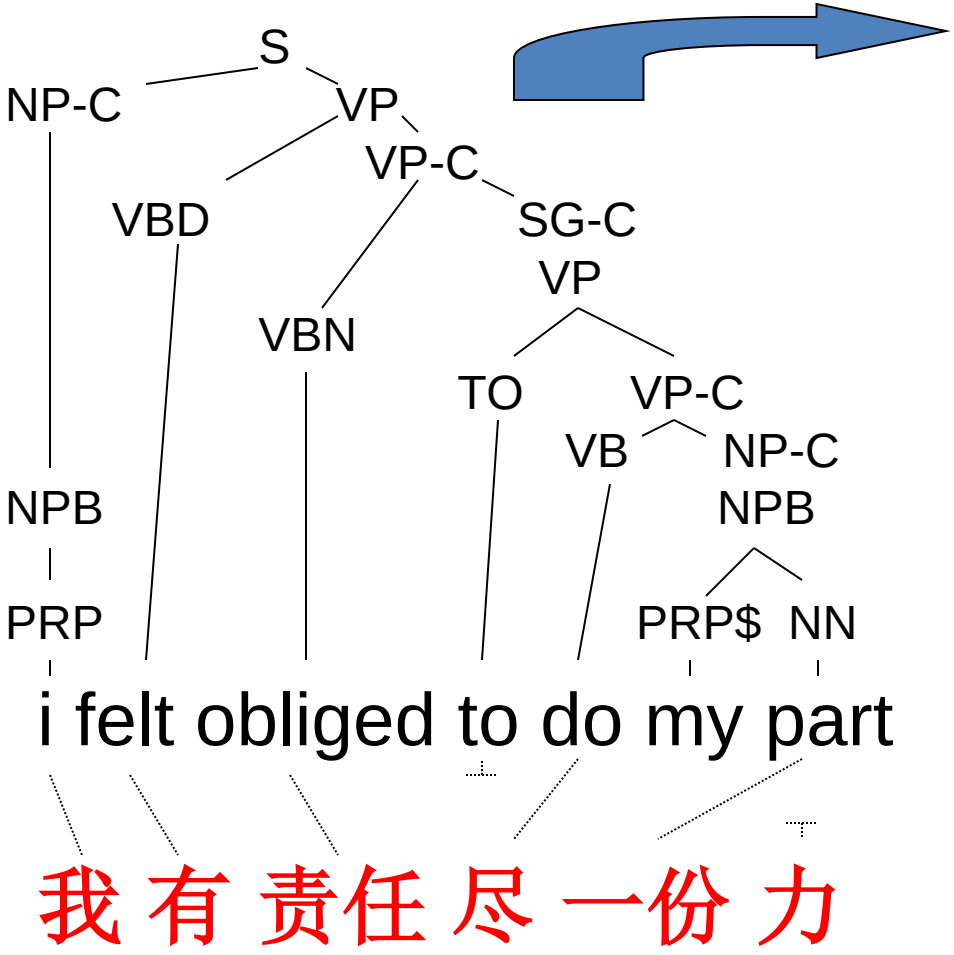
Original input:



Final output:

kare **,** ***wa*** **,** ***ongaku*** **,** ***o*** **,** ***kiku*** **,** ***no*** **,** ***ga*** **,** ***daisuki*** **,** ***desu***

Tree Transducers Can be Extracted from Bilingual Data (Galley et al, 04)



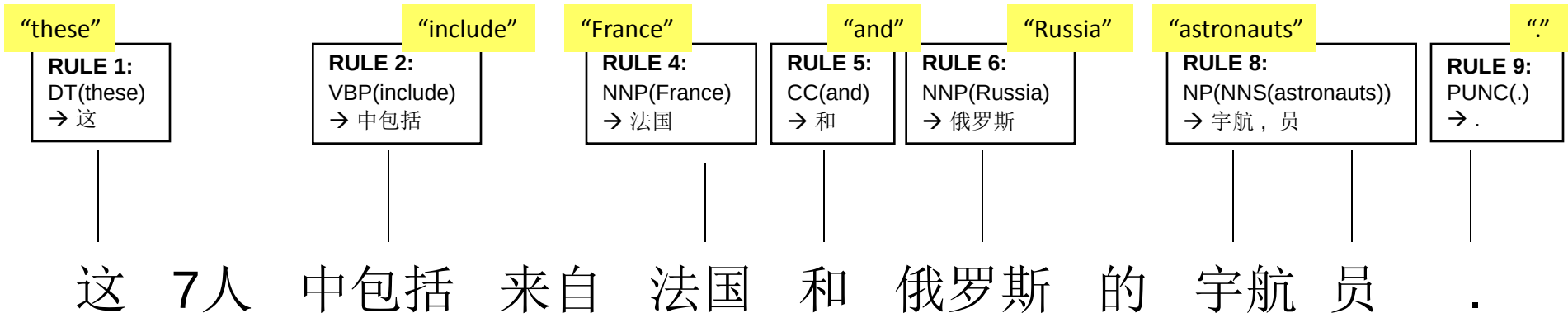
TREE TRANSDUCER RULES:

VBD(felt) → 有
VBN(obliged) → 责任
VB(do) → 尽
NN(part) → 一份
NN(part) → 一份 力
VP-C(x0:VBN x1:SG-C) → x0 x1
VP(TO(to) x0:VP-C) → x0
…
S(x0:NP-C x1:VP) → x0 x1

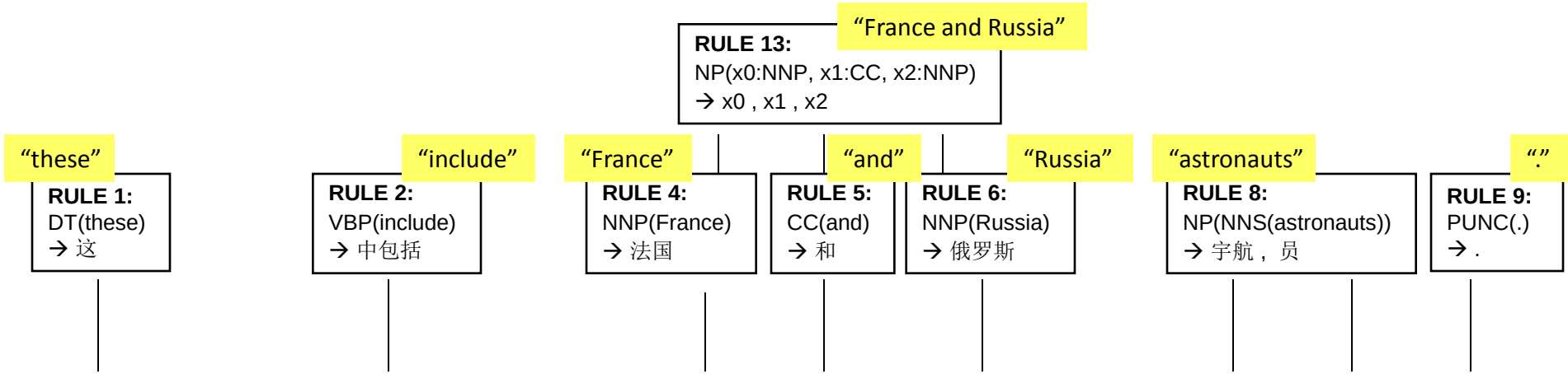
Syntax-Based Decoding

这 7人 中包括 来自 法国 和 俄罗斯 的 宇航 员 .

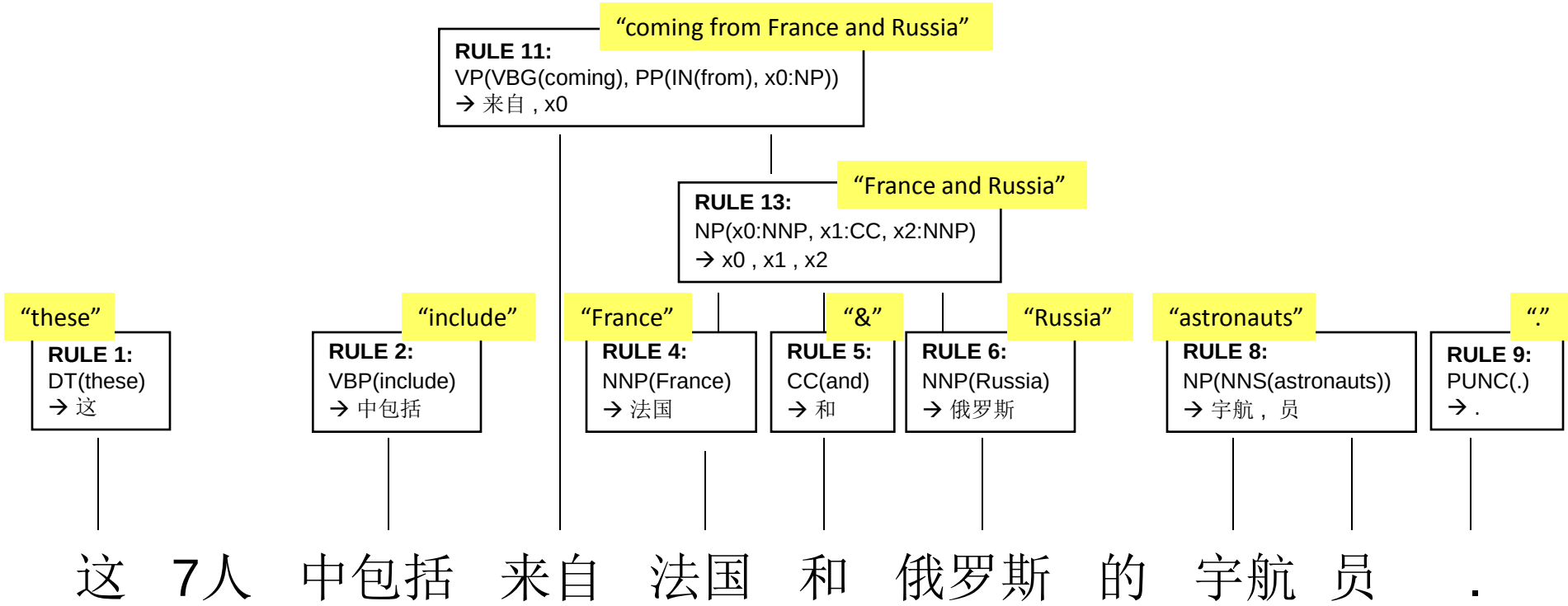
Syntax-Based Decoding



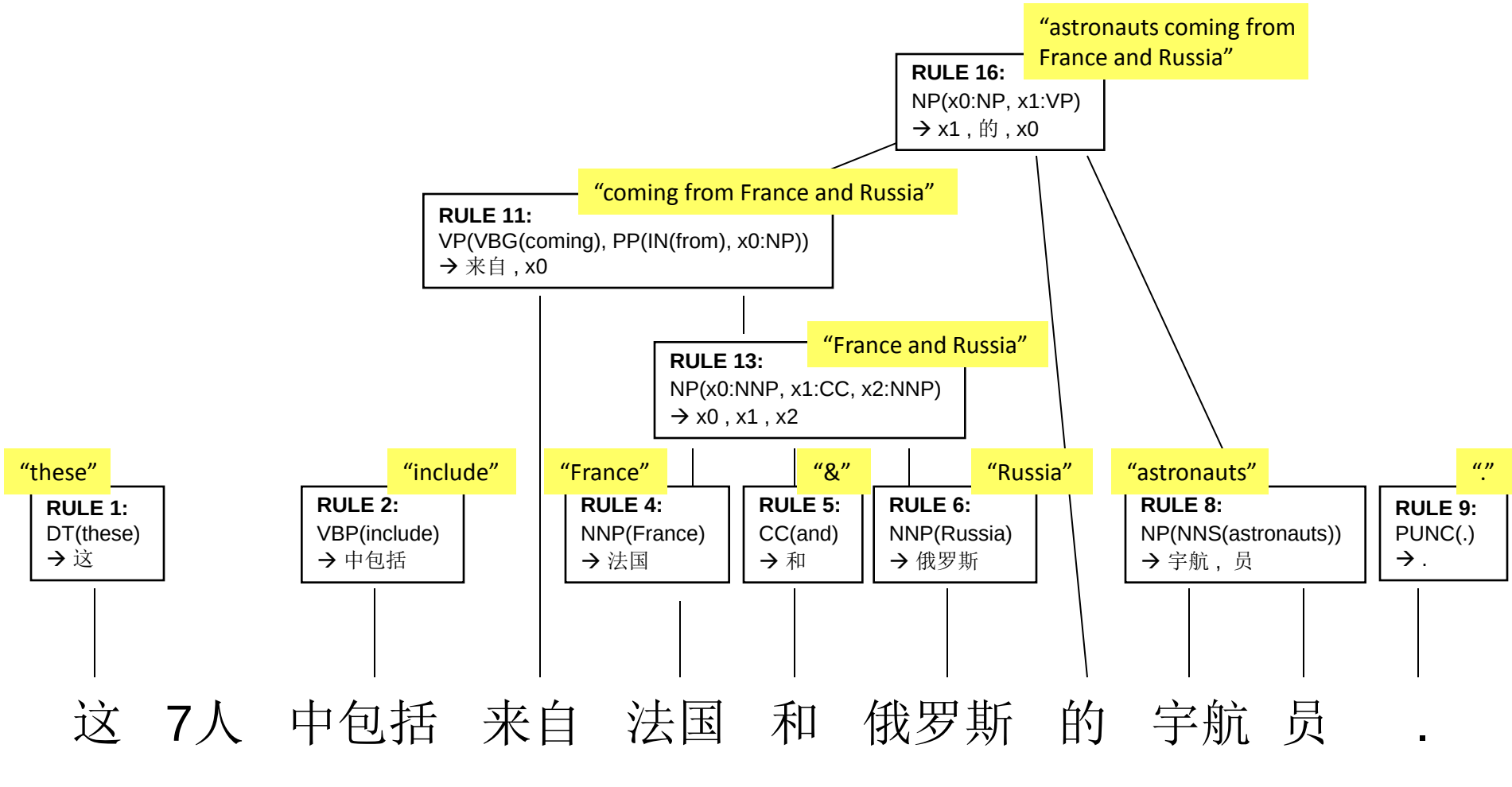
Syntax-Based Decoding

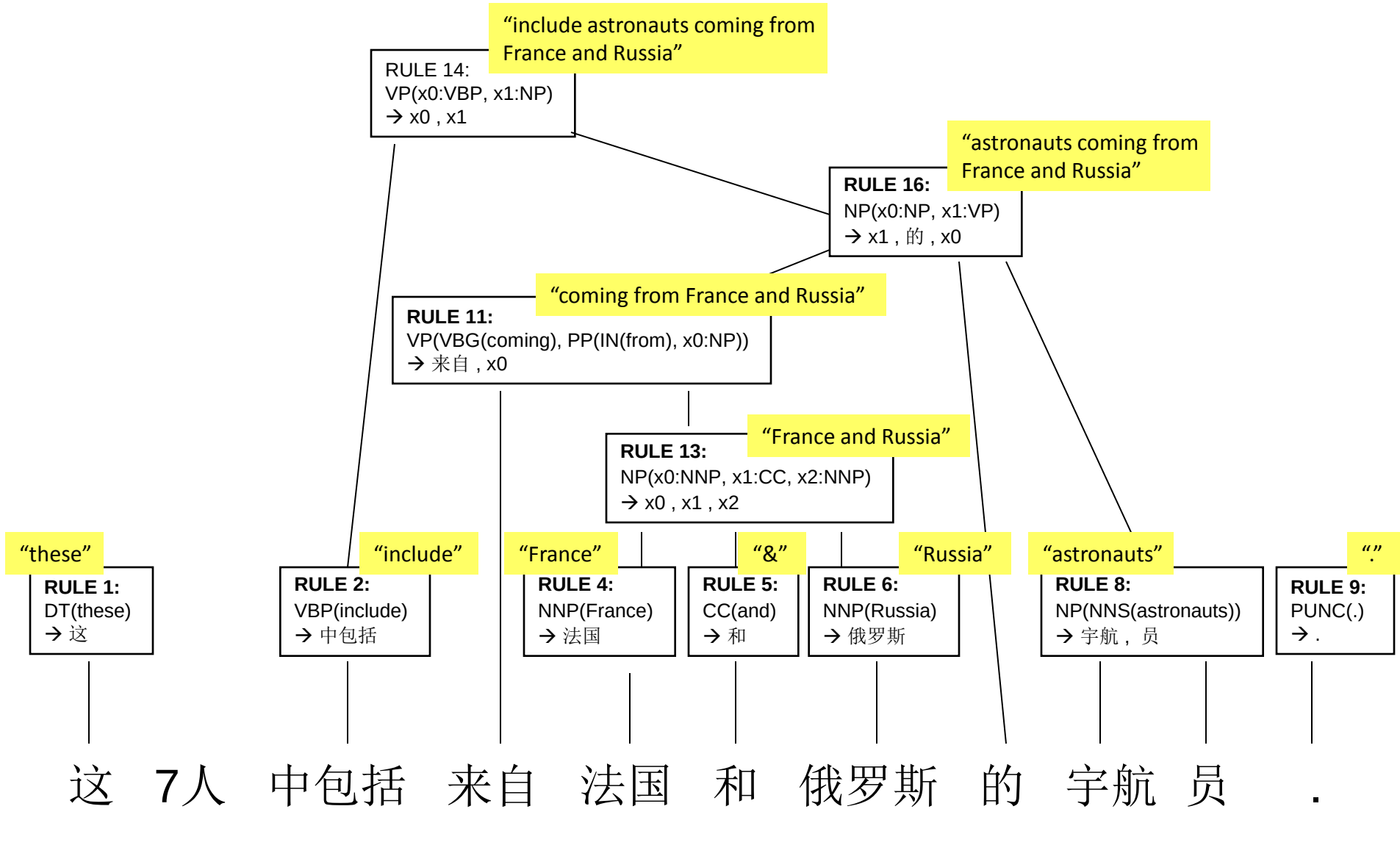


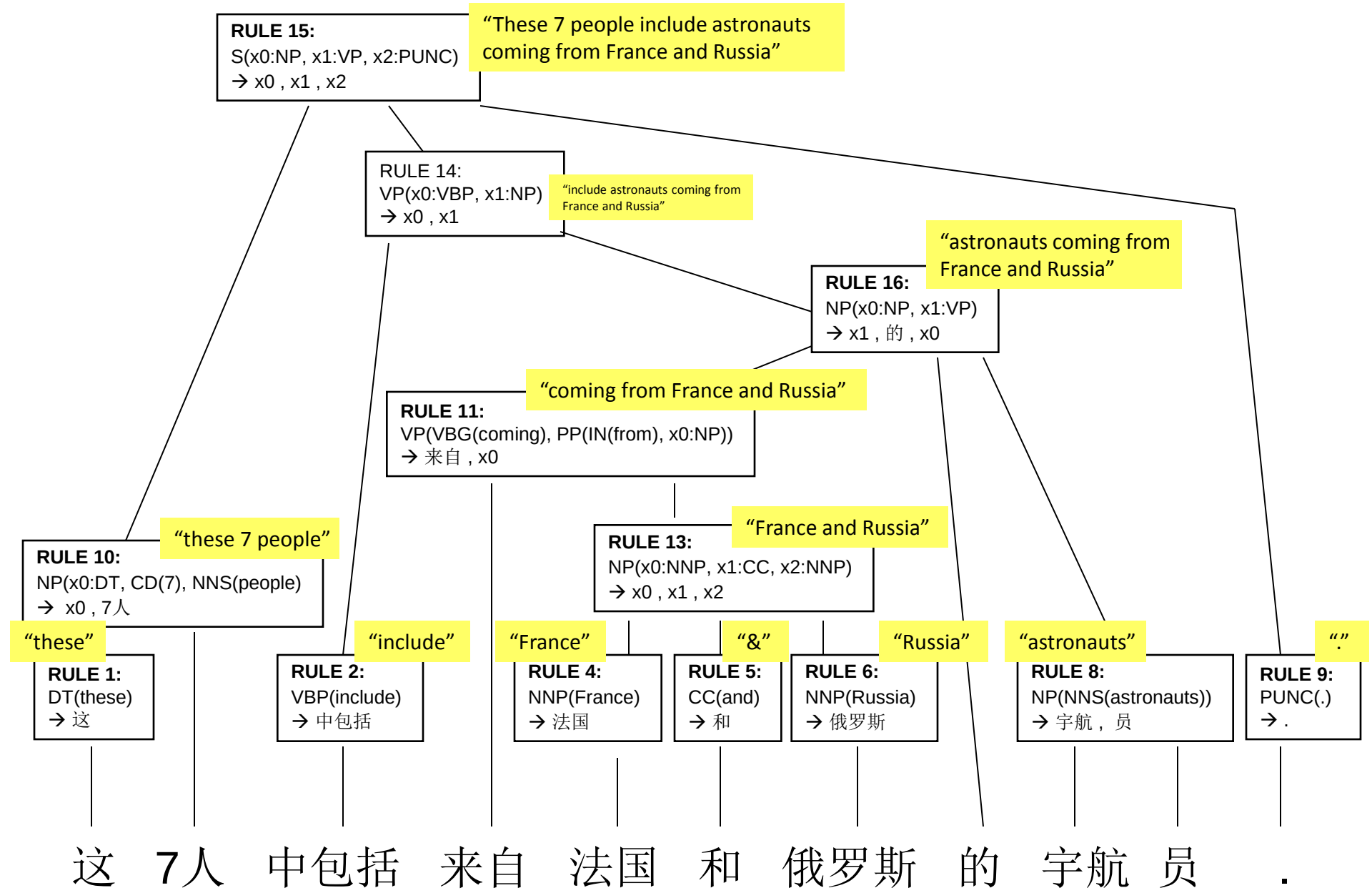
Syntax-Based Decoding



Syntax-Based Decoding




"include astronauts coming from France and Russia"
RULE 14:
VP(x0:VBP, x1:NP)
→ x0 , x1
"astronauts coming from France and Russia"
RULE 16:
NP(x0:NP, x1:VP)
→ x1 , 的 , x0
"coming from France and Russia"
RULE 11:
VP(VBG(coming), PP(IN(from), x0:NP))
→ 来自 , x0
"France and Russia"
RULE 13:
NP(x0:NNP, x1:CC, x2:NNP)
→ x0 , x1 , x2
"these"
RULE 1:
DT(these)
→ 这
"include"
RULE 2:
VBP(include)
→ 中包括
"France"
RULE 4:
NNP(France)
→ 法国
"&"
RULE 5:
CC(and)
→ 和
"Russia"
RULE 6:
NNP(Russia)
→ 俄罗斯
"astronauts"
RULE 8:
NP(NNS(astronauts))
→ 宇航 , 员
"."
RULE 9:
PUNC(.)
→ .
这 7人 中包括 来自 法国 和 俄罗斯 的 宇航 员 .


RULE 15:
S(x0:NP, x1:VP, x2:PUNC)
→ x0 , x1 , x2
“These 7 people include astronauts coming from France and Russia”
RULE 14:
VP(x0:VBP, x1:NP)
→ x0 , x1
“include astronauts coming from France and Russia”
RULE 16:
NP(x0:NP, x1:VP)
→ x1 , 的 , x0
“astronauts coming from France and Russia”
RULE 11:
VP(VBG(coming), PP(IN(from), x0:NP))
→ 来自 , x0
“coming from France and Russia”
RULE 13:
NP(x0:NNP, x1:CC, x2:NNP)
→ x0 , x1 , x2
“France and Russia”
RULE 10:
NP(x0:DT, CD(7), NNS(people)
→ x0 , 7人
“these 7 people”
RULE 1:
DT(these)
→ 这
“these”
RULE 2:
VBP(include)
→ 中包括
“include”
RULE 4:
NNP(France)
→ 法国
“France”
RULE 5:
CC(and)
→ 和
“&”
RULE 6:
NNP(Russia)
→ 俄罗斯
“Russia”
RULE 8:
NP(NNS(astronauts))
→ 宇航 , 员
“astronauts”
RULE 9:
PUNC(.)
→ .
“.”
这 7人 中包括 来自 法国 和 俄罗斯 的 宇航 员 .

General-Purpose Algorithms for Tree Automata

	String Automata Algorithms	Tree Automata Algorithms
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)
EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)
Determinization …	… of weighted string acceptors (Mohri, 1997)	… of weighted tree acceptors (Borchardt & Vogler, 2003; May & Knight, 2005)
Intersection	WFSA intersection	Tree acceptor intersection
Applying transducers	string → WFST → WFSA	tree → TT → weighted tree acceptor
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)
General-purpose tools	Carmel, OpenFST	Tiburon (May & Knight 10)

Machine Translation

Phrase-based MT

source string → target string

Syntax-based MT

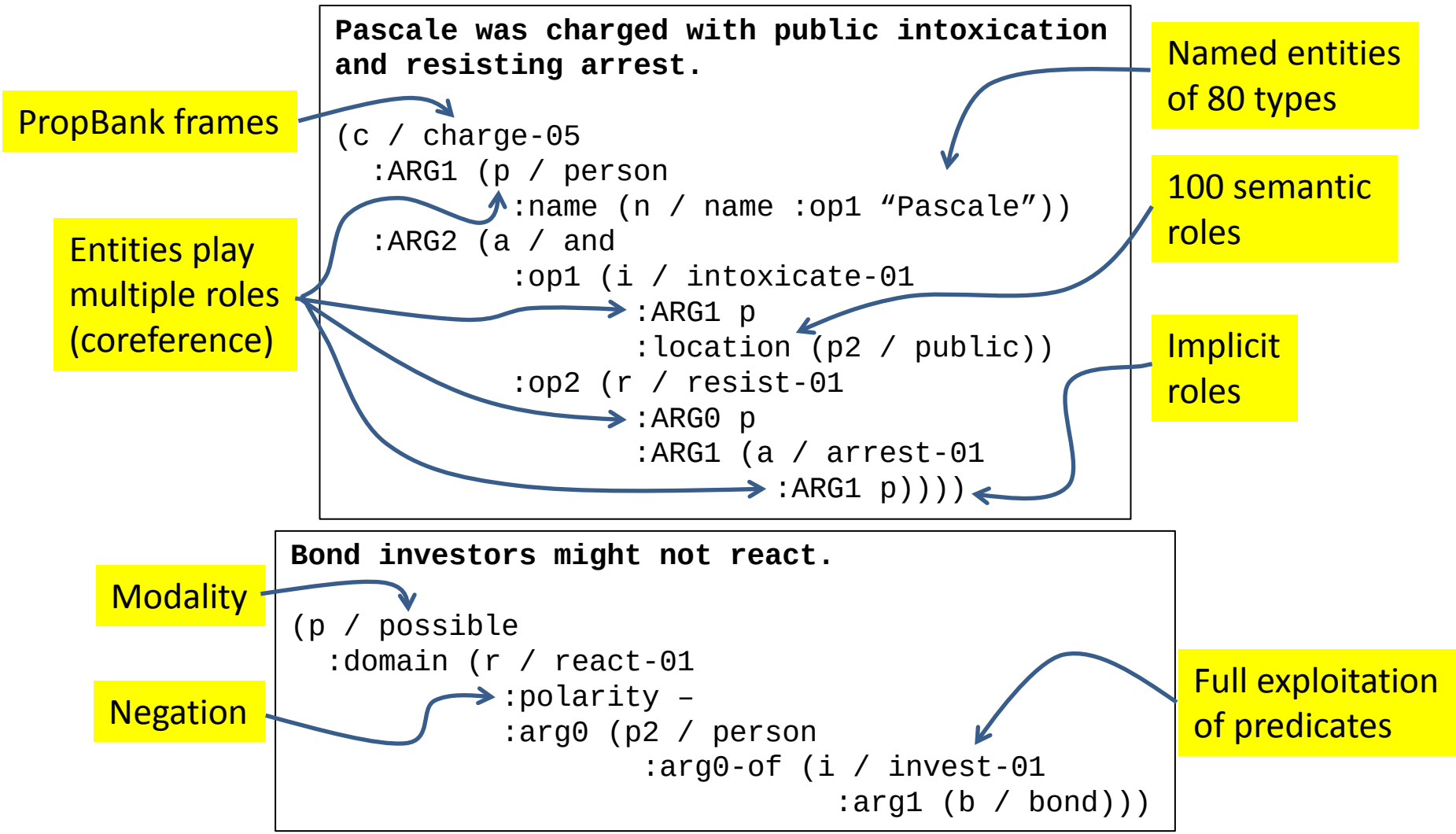
source string → source tree → target tree → target string

Meaning-based MT

source string → source tree → meaning representation → target tree → target string

graphs

Abstract Meaning Representation (AMR)

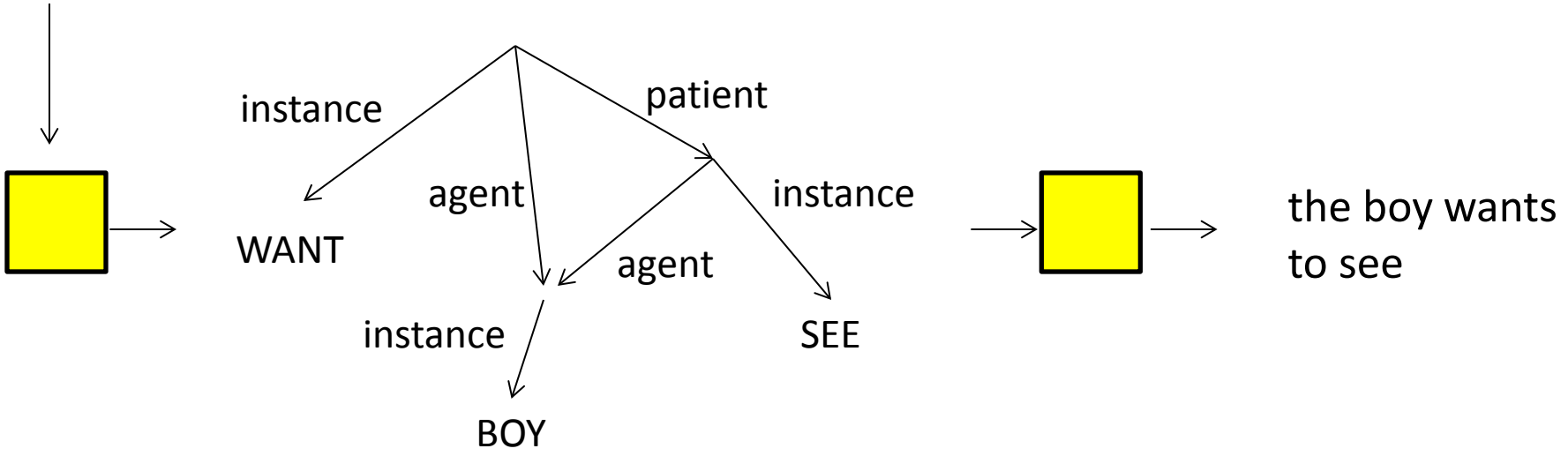


General-Purpose Algorithms for Feature Structures (Graphs)

	String Automata Algorithms	Tree Automata Algorithms	Graph Automata Algorithms?
N-best answer extraction	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)	
Unsupervised EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)	
Determinization, minimization	… of weighted string acceptors (Mohri, 1997)	… of weighted tree acceptors (Borchardt & Vogler, 2003; May & Knight, 2005)	
Intersection	WFSA intersection	Tree acceptor intersection	
Application of transducers	string → WFST → WFSA	tree → TT → weighted tree acceptor	
Composition of transducers	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)	
Software tools	Carmel, OpenFST	Tiburon (May & Knight 10)	

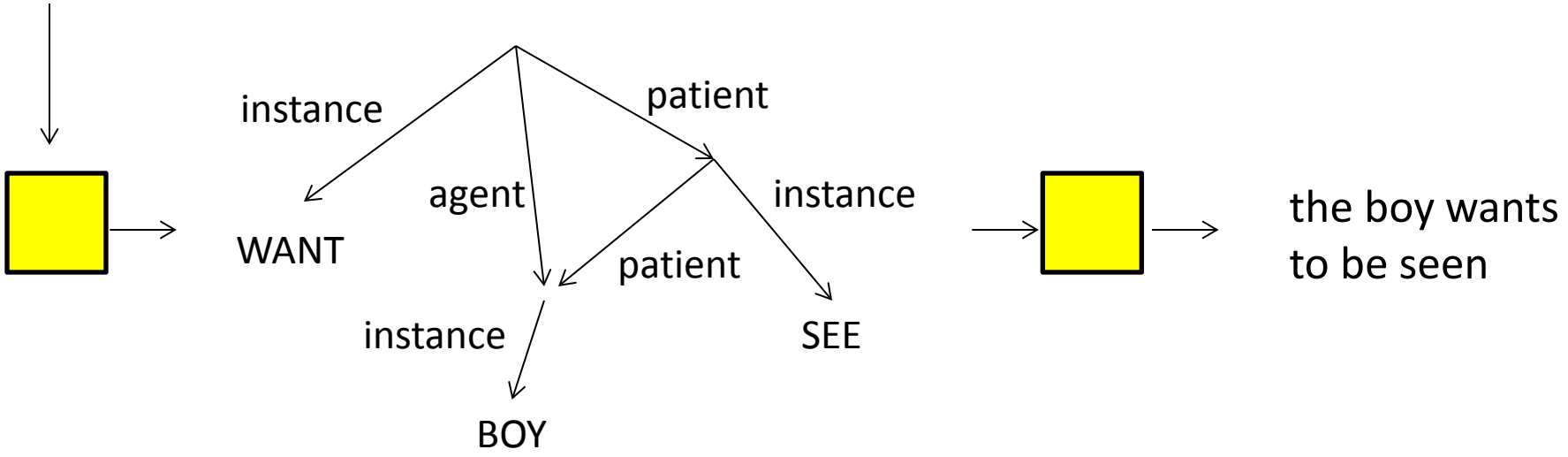
Mapping Between Meaning and Text

Umuhungu arashaka kubona.



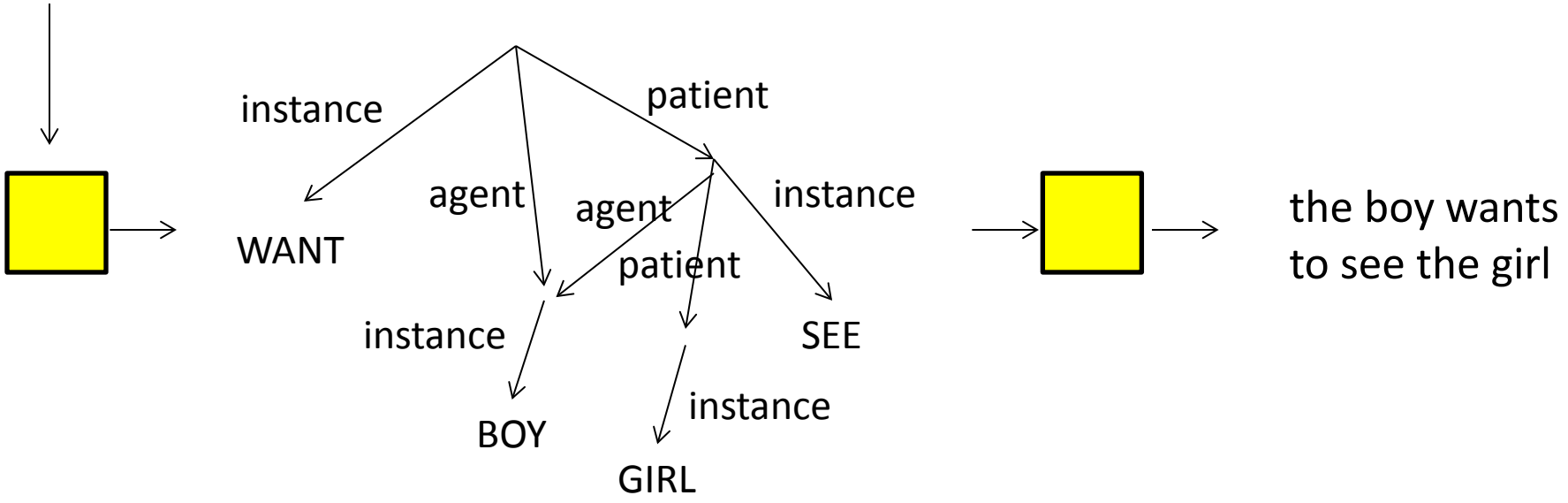
Mapping Between Meaning and Text

Umuhungu arashaka kubonwa.



Mapping Between Meaning and Text

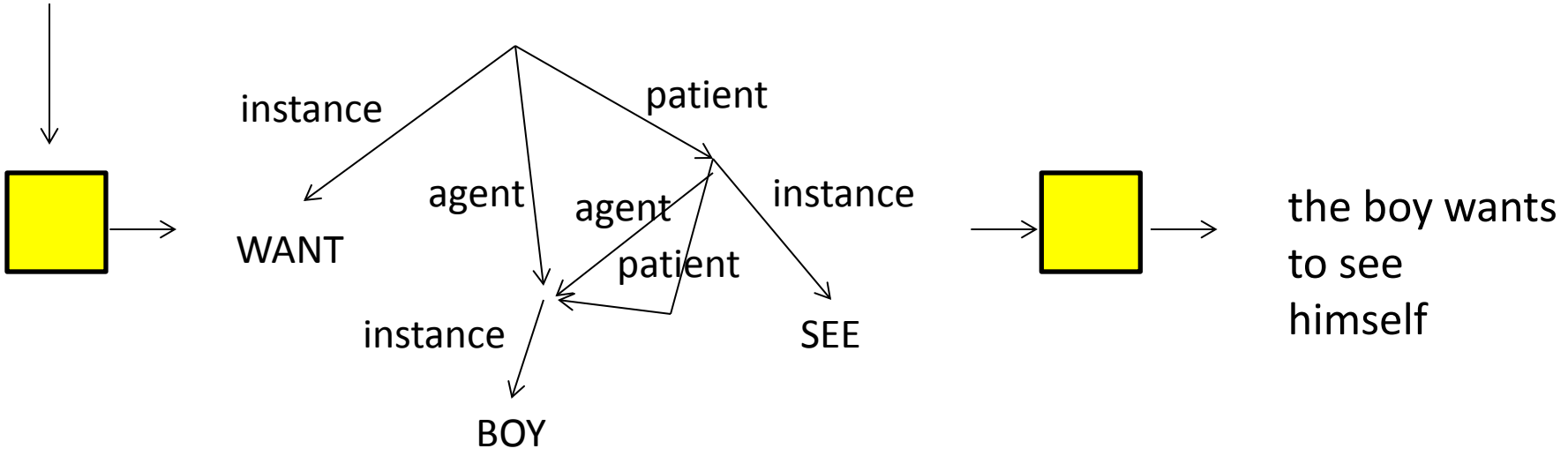
Umuhungu arashaka
kubona umukobwa



the boy wants
to see the girl

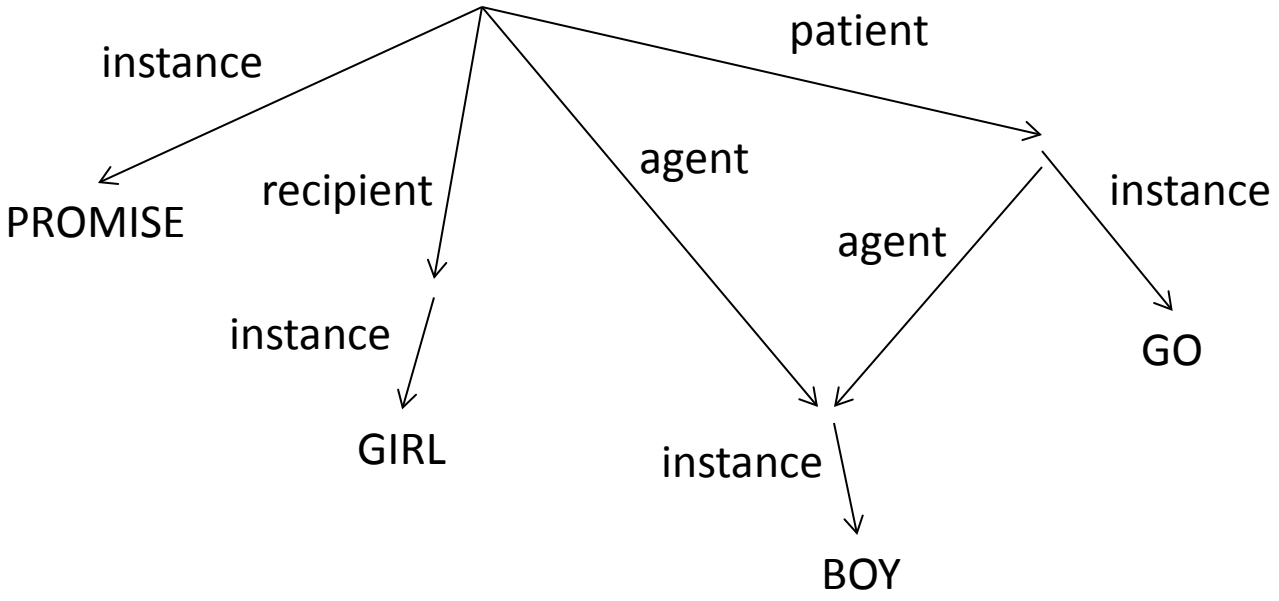
Mapping Between Meaning and Text

Umuhungu arashaka kwibona.



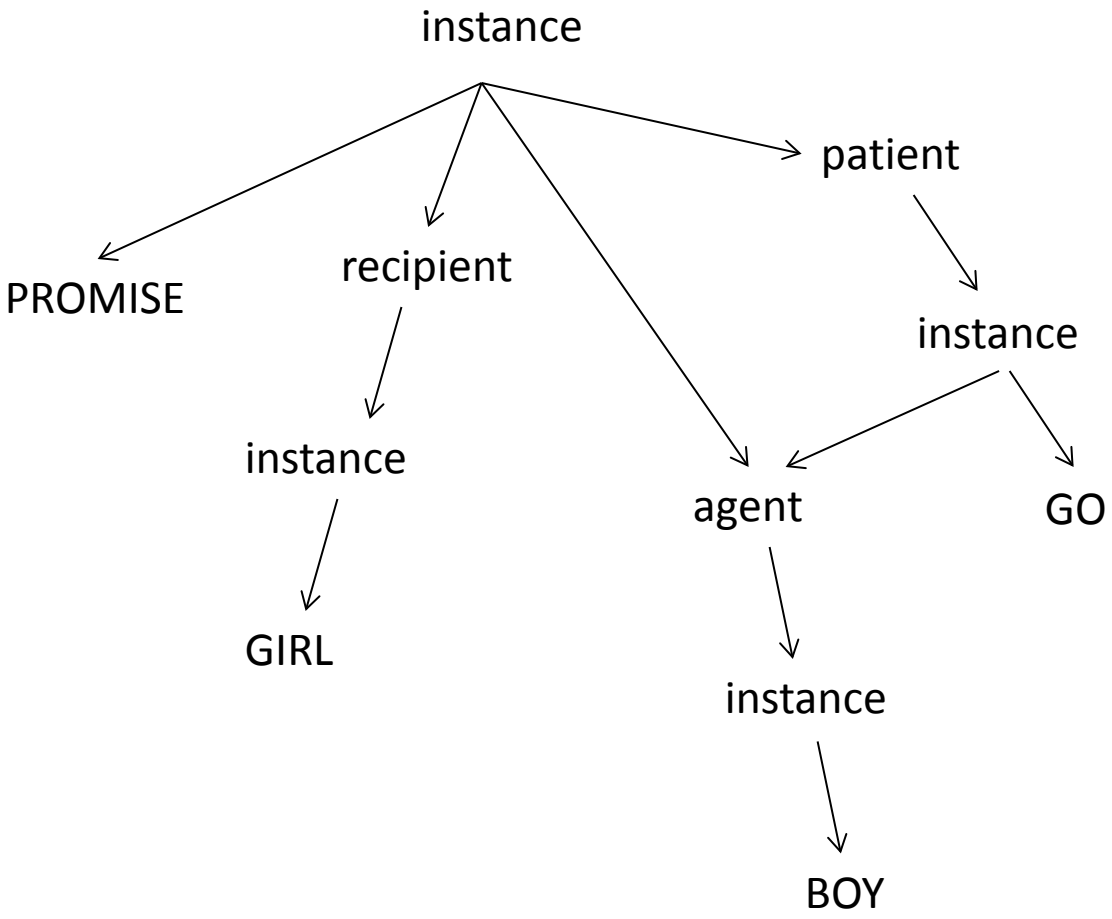
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



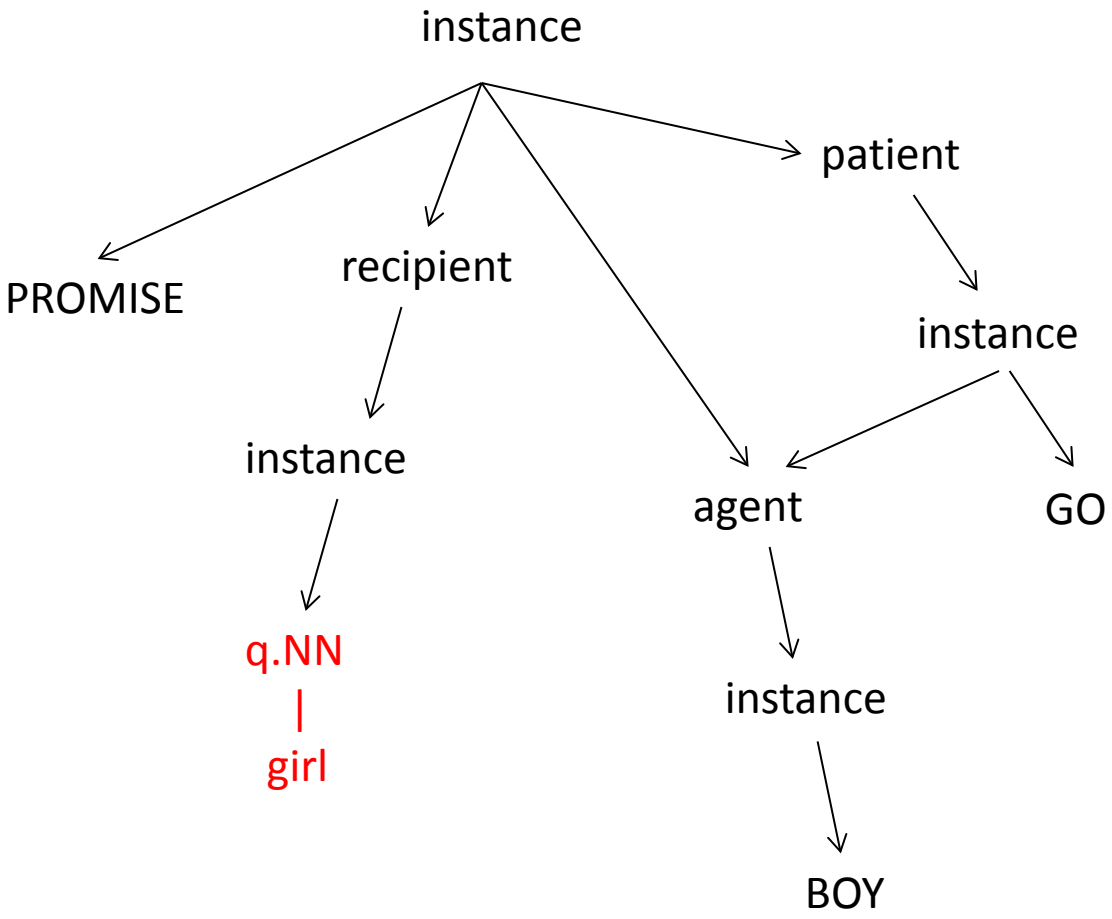
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



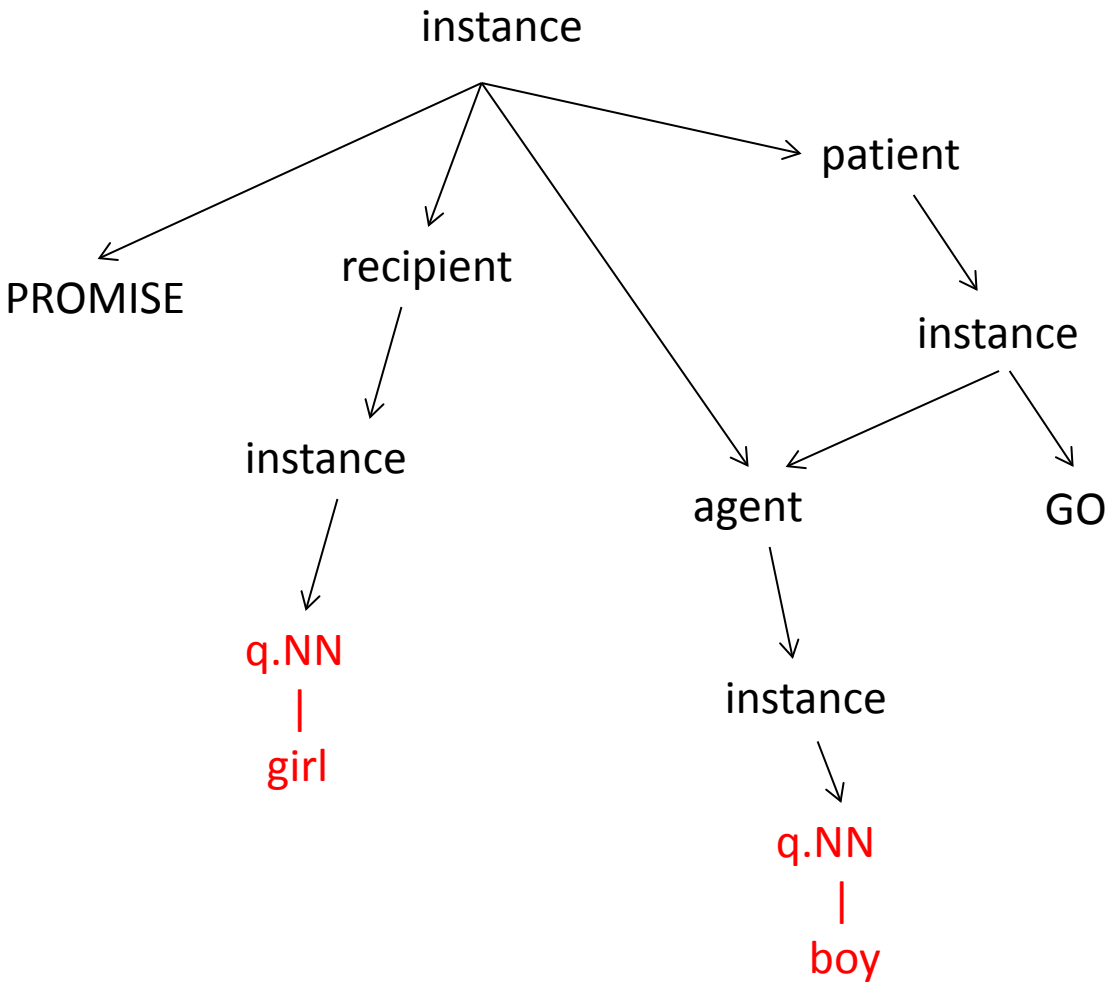
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



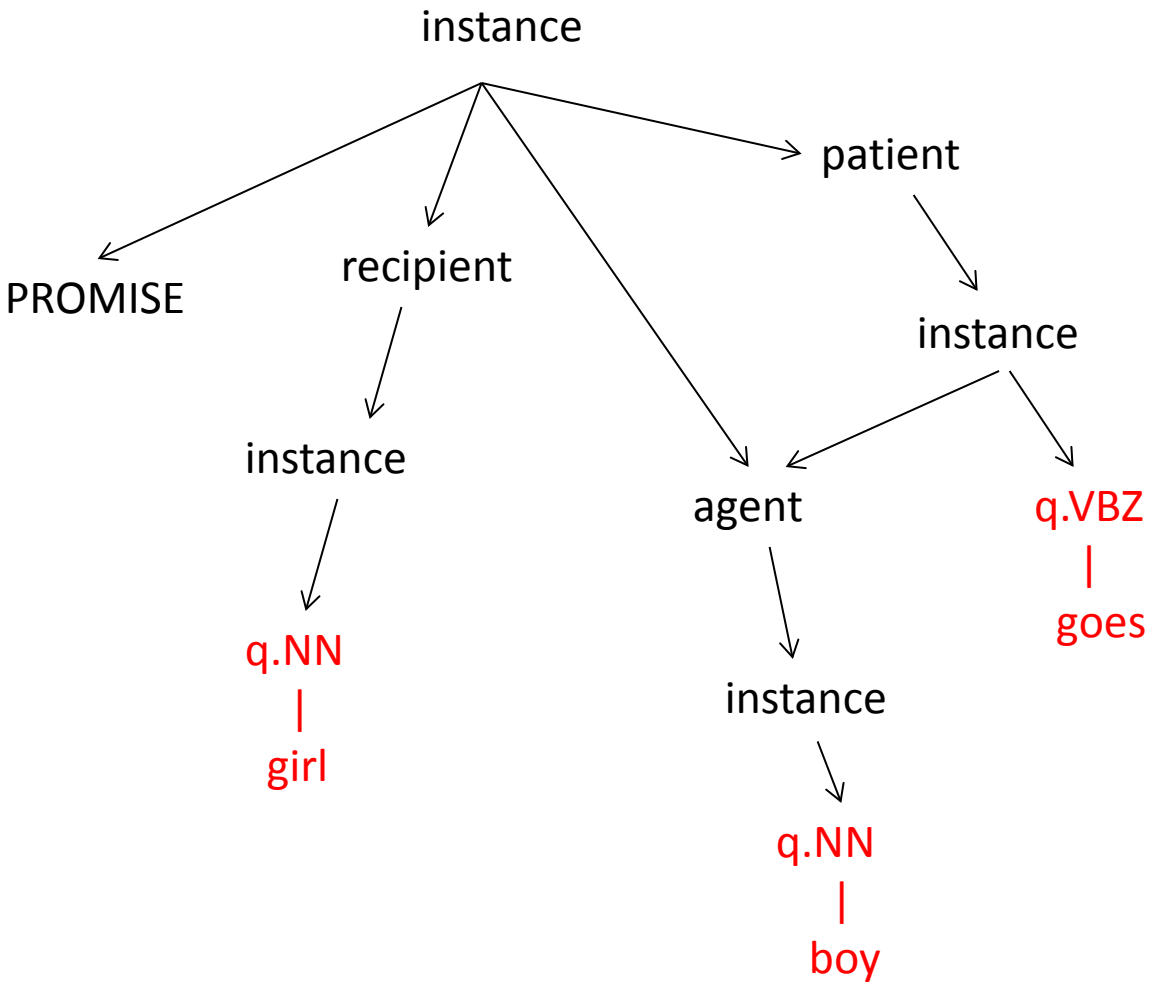
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



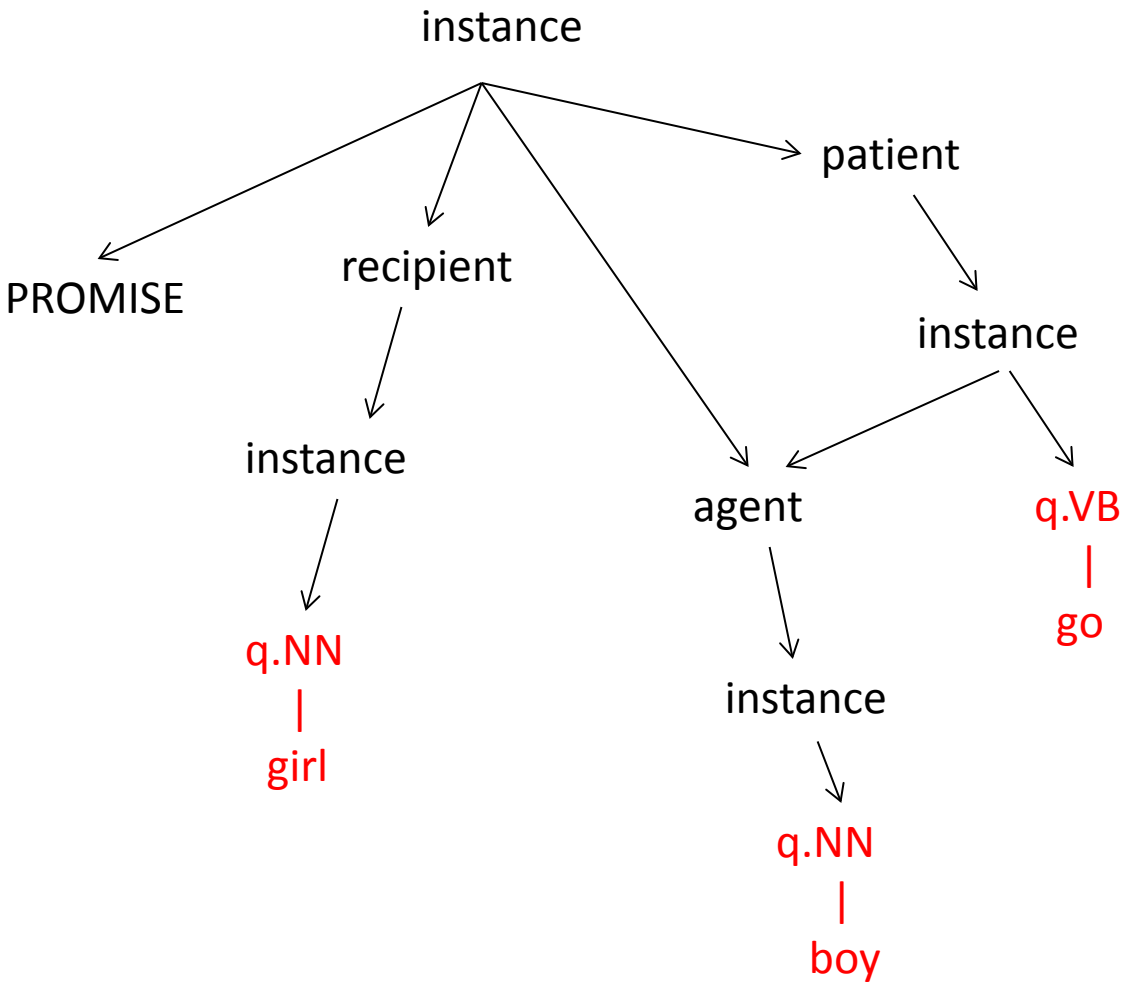
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



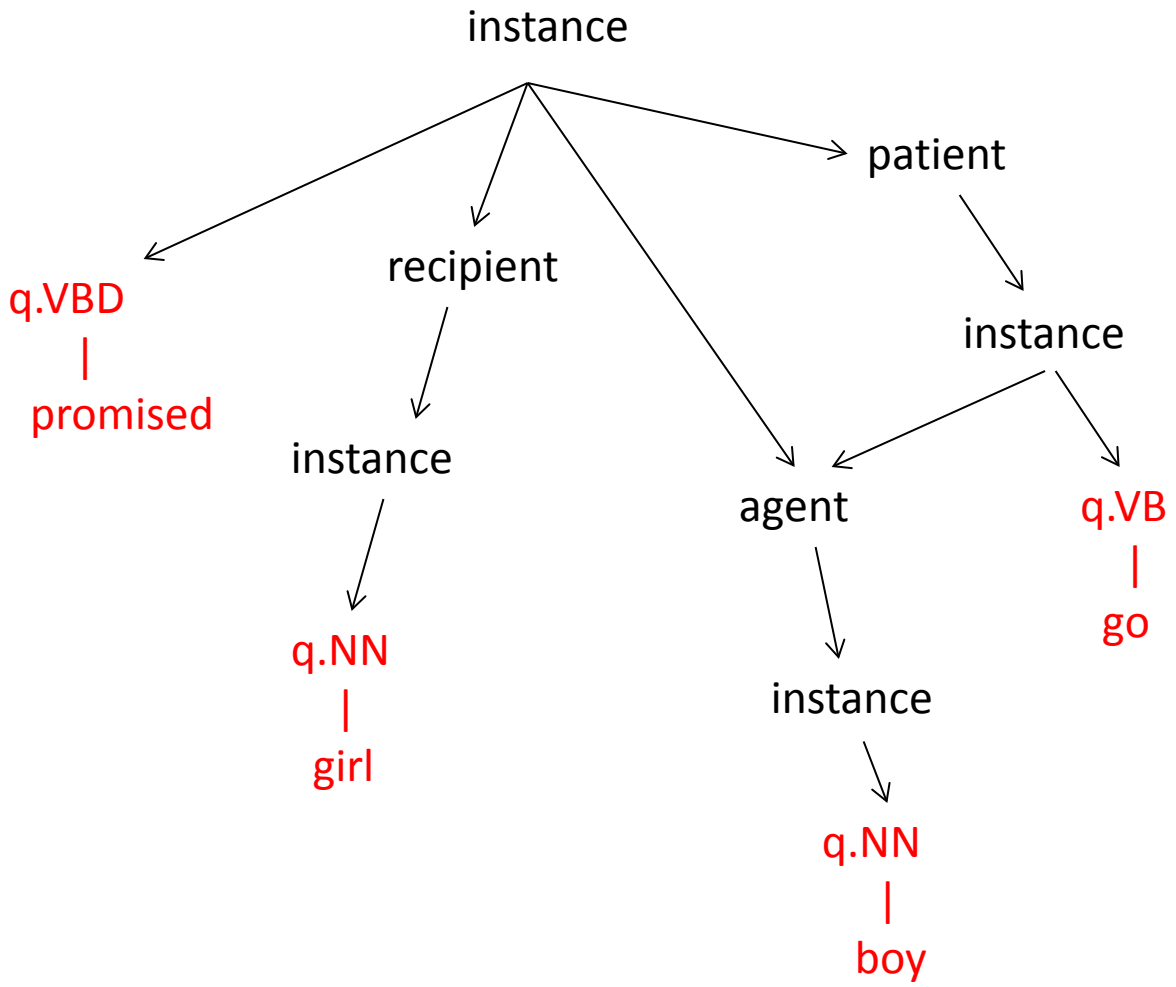
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



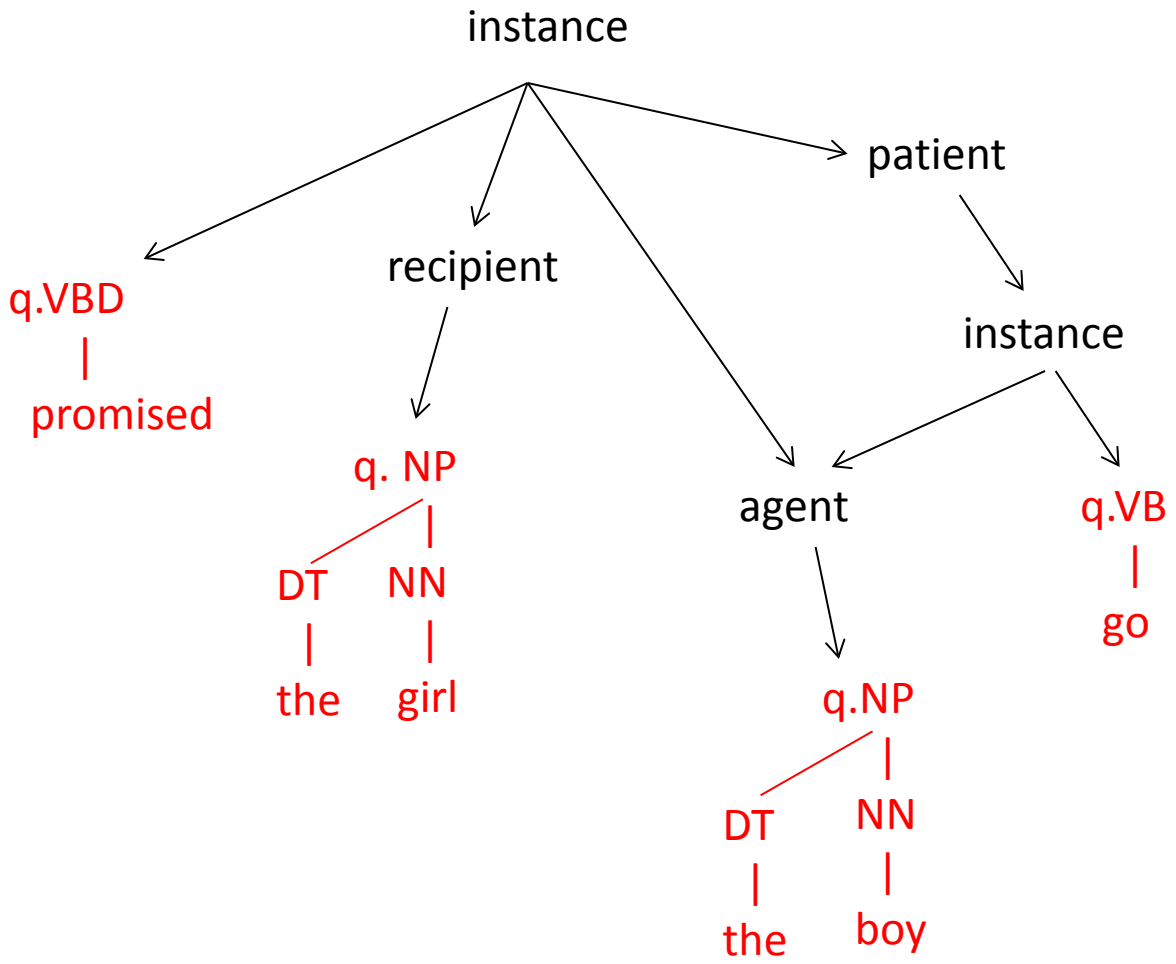
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



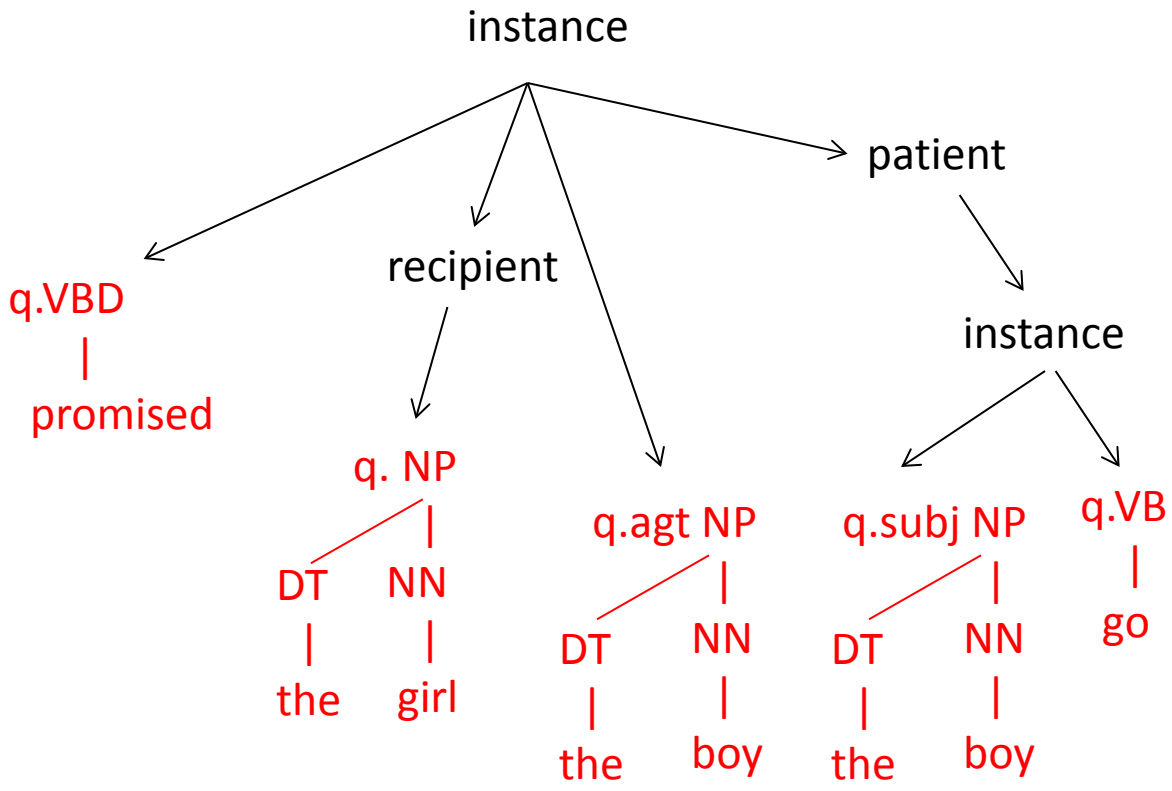
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



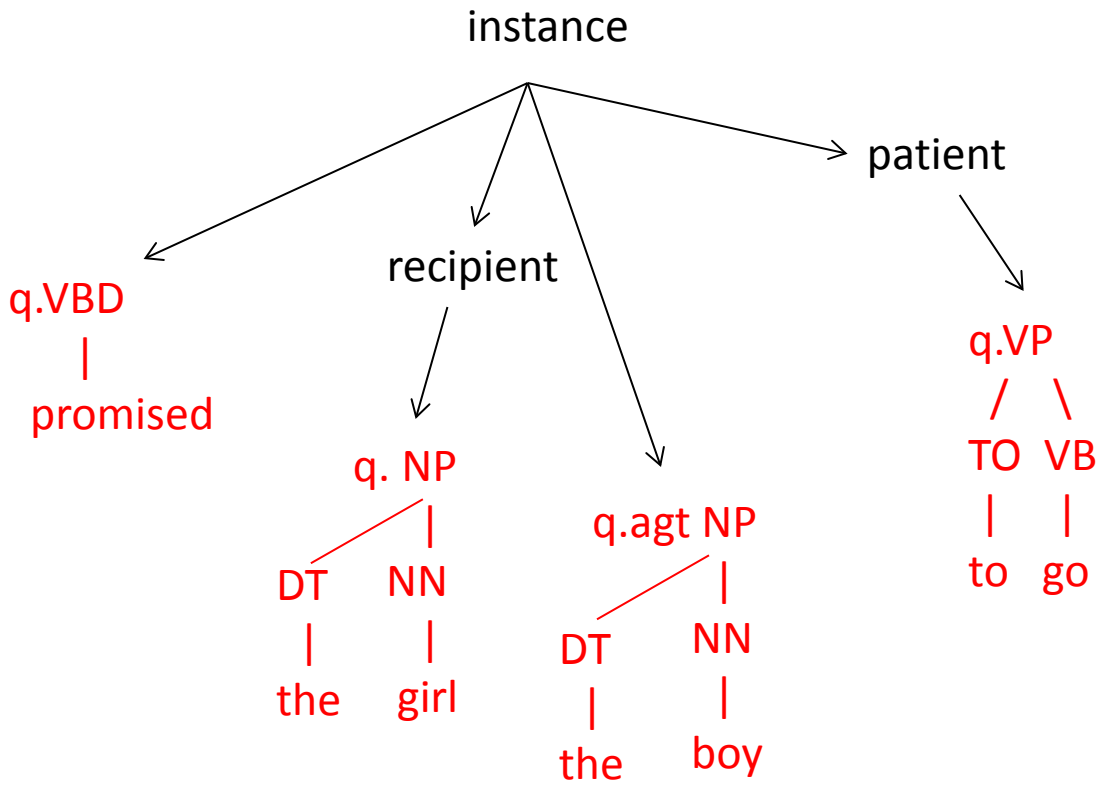
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



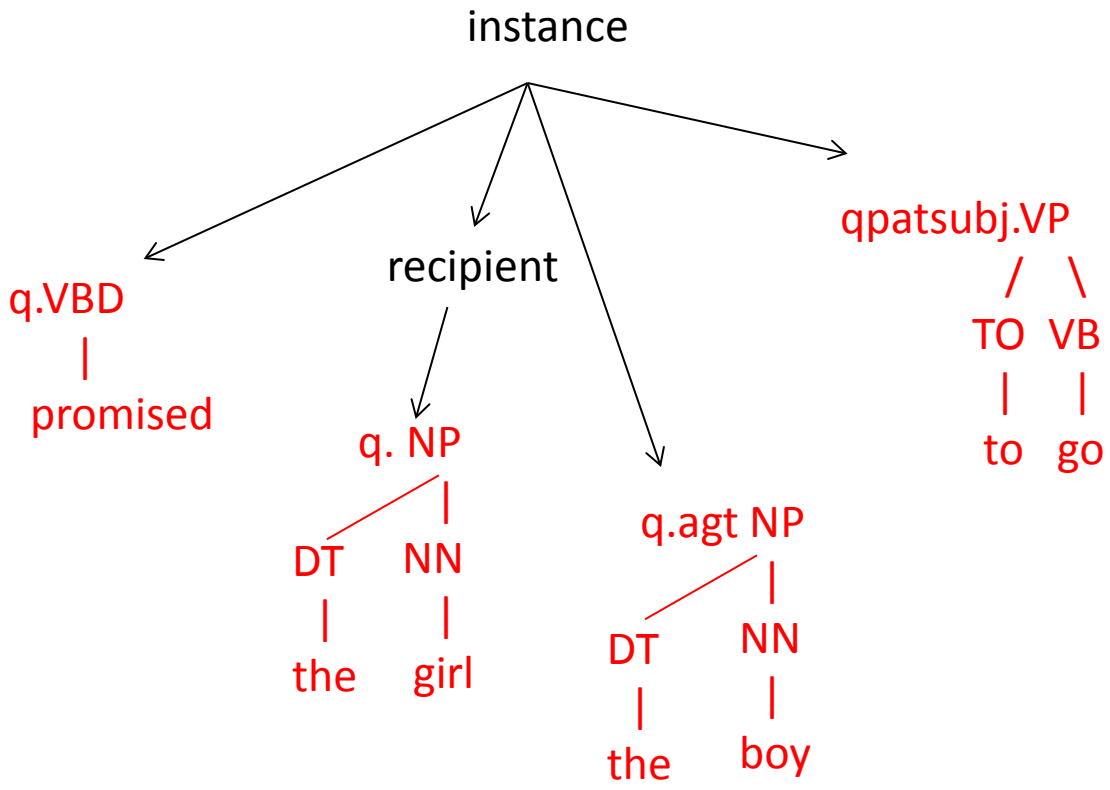
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



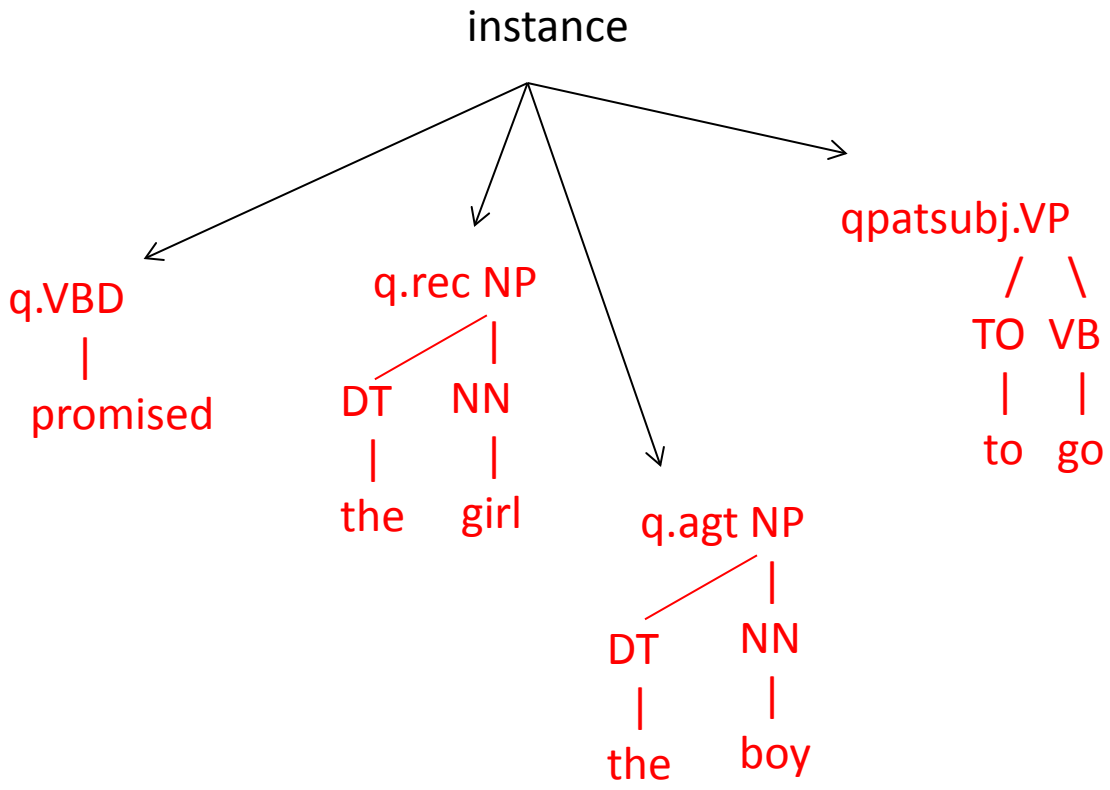
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



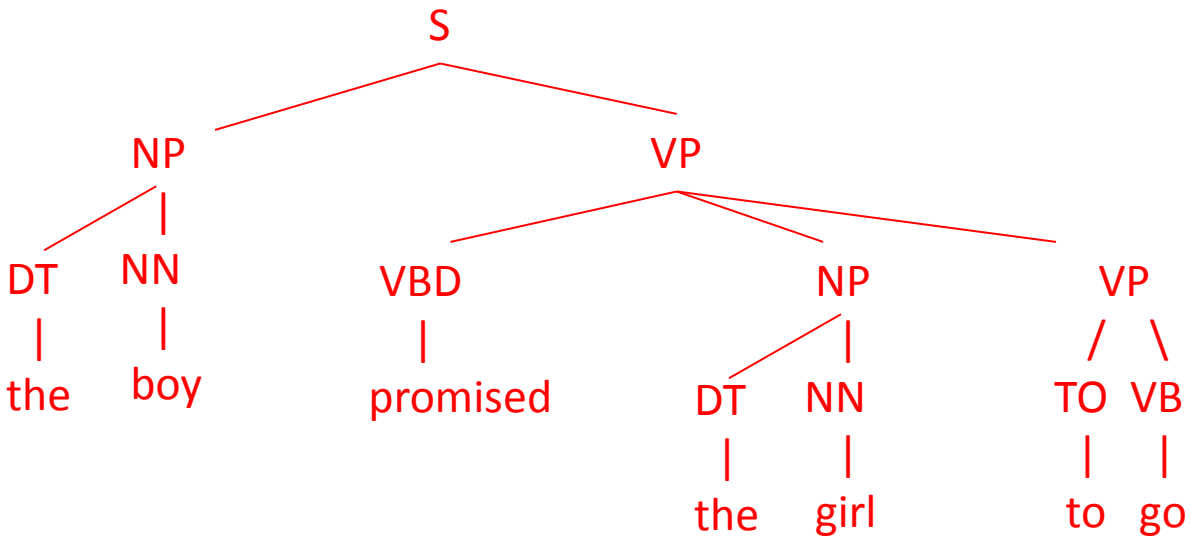
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



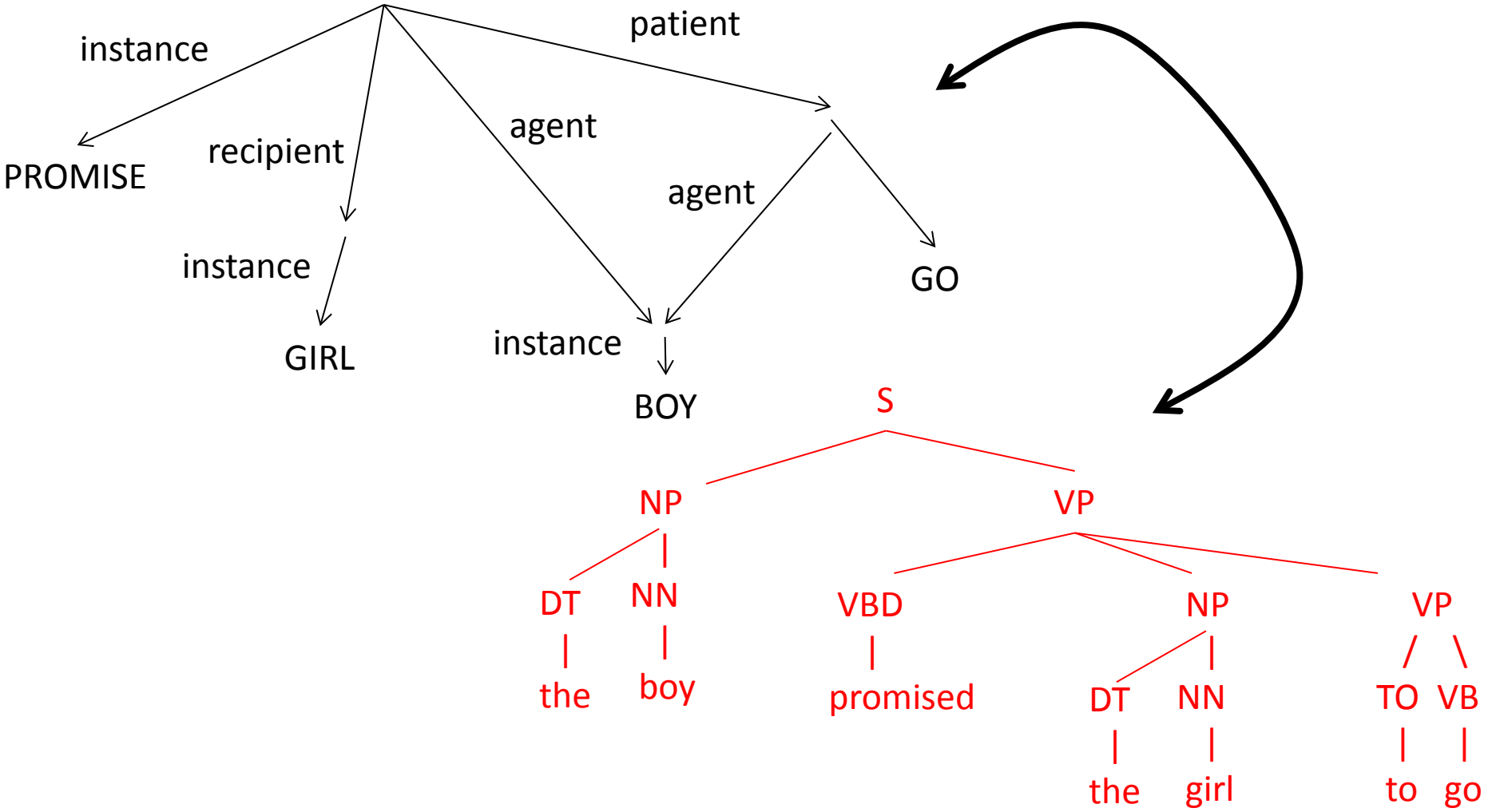
Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



Bottom-Up DAG-to-Tree Transduction

(Kamimura & Slutski 82, Quernheim & Knight 2012ab)



Some Problems

- Non-local
- Hard to probabilize, train
- Computational properties difficult to work out

Further Developments

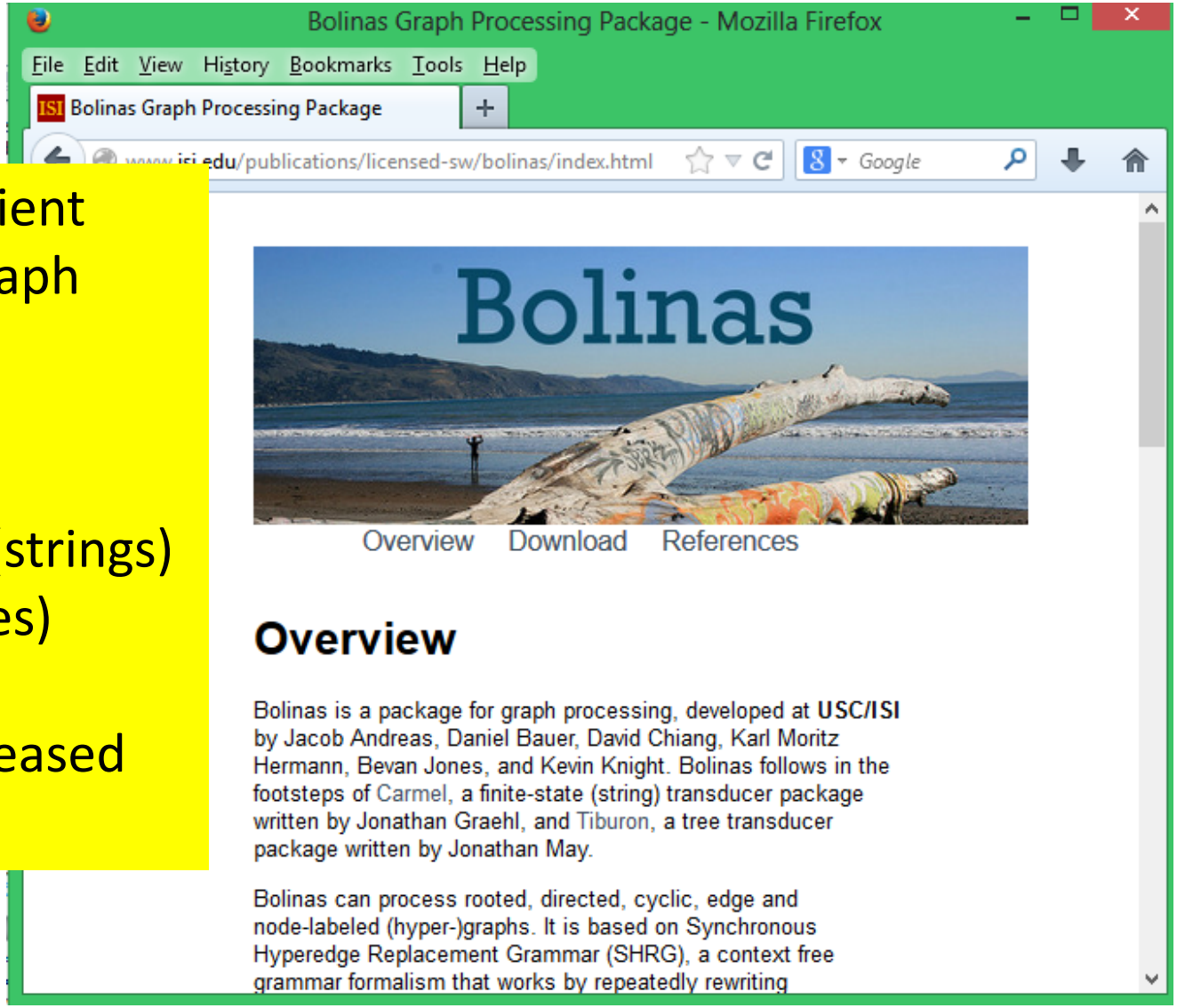
- Hyperedge Replacement Grammar (HRG)
  - synchronous version for transduction
  - proof-of-concept MT system [COLING 2012]
- Novel algorithm for graph parsing [ACL 2013]
- Map of theoretical and computational properties
  - closure, complexity [subm. CIAA 2014]
- Empirical fitness to linguistic data [subm. LREC 2014]

Bolinas Graph Processing Toolkit

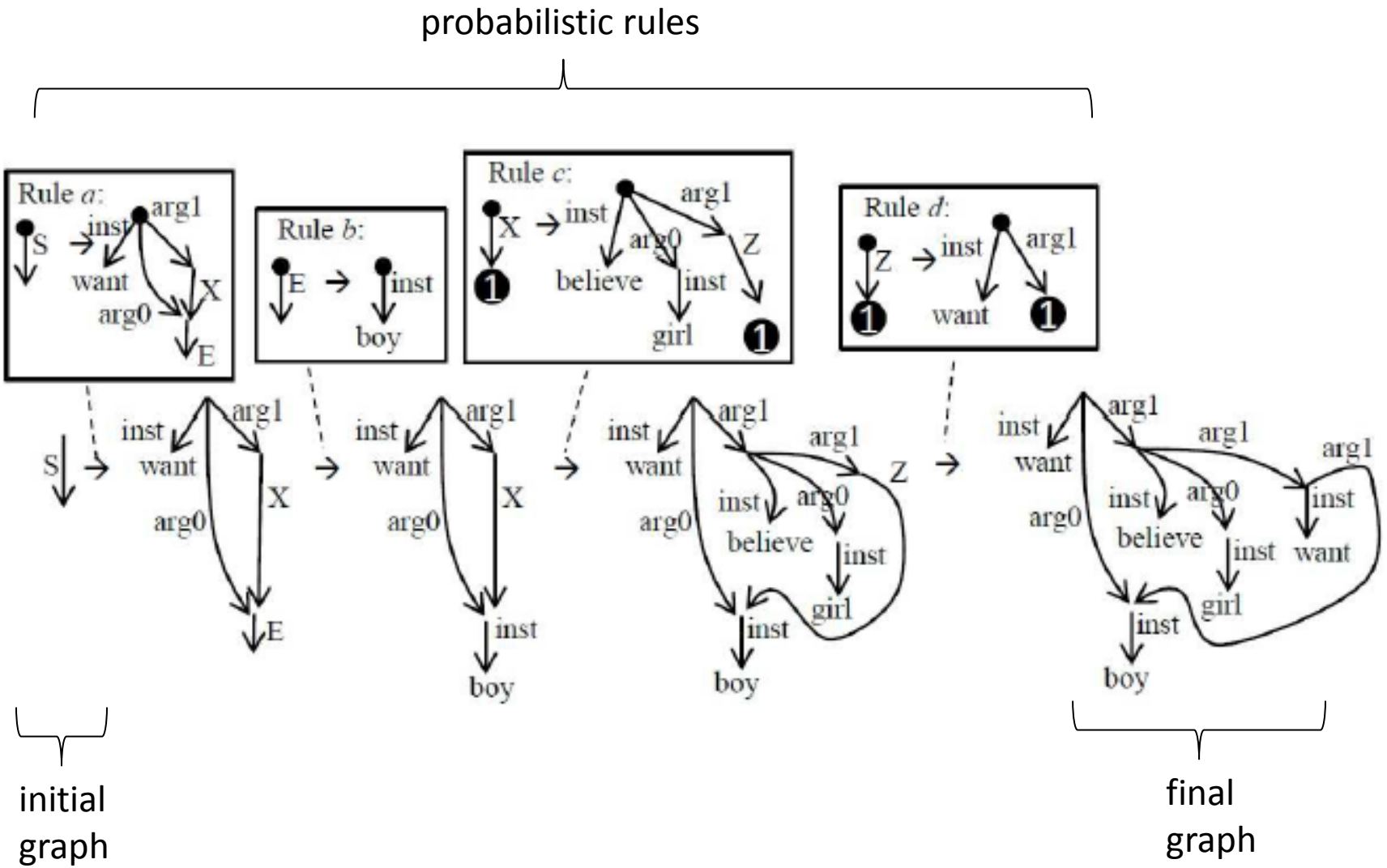
Implements efficient operations for graph acceptors and transducers

Sister to Carmel (strings) and Tiburon (trees)

Documented/released August 2013

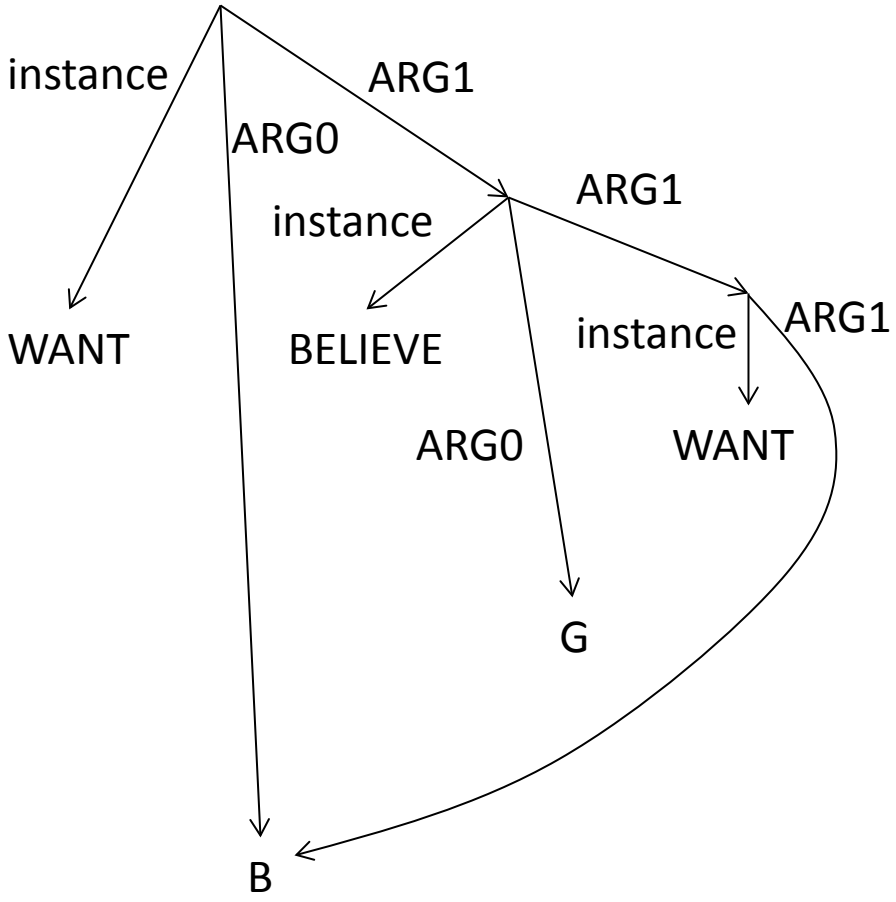


How an HRG Generates a Graph



HRG Derivation 1

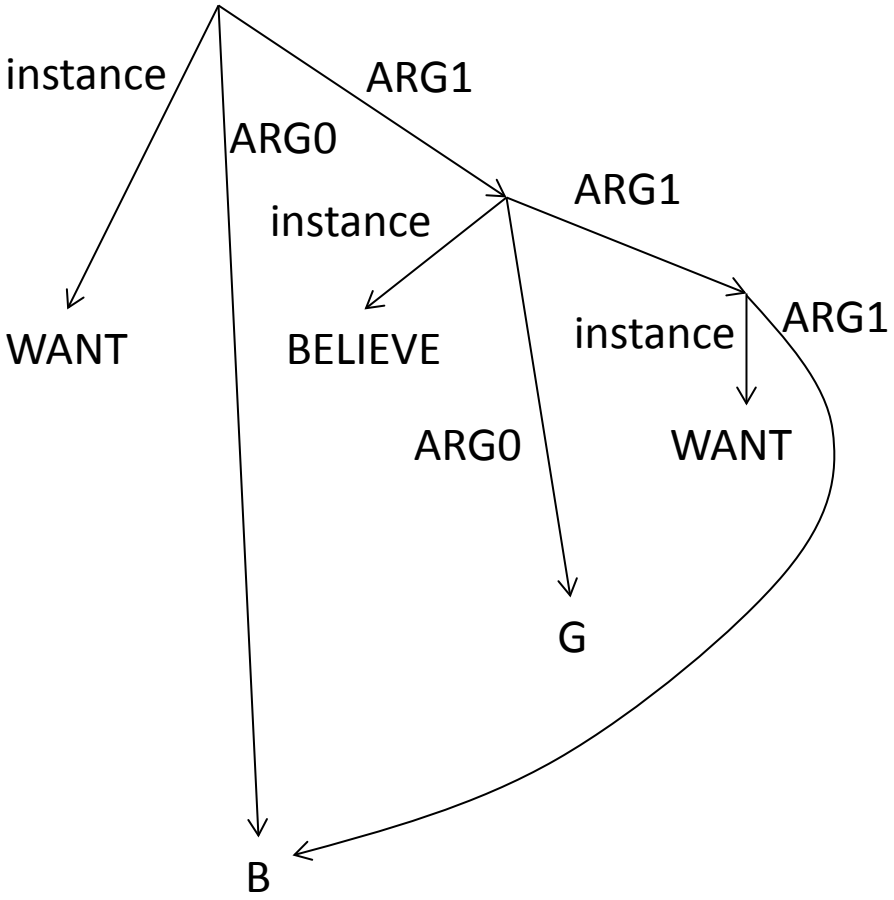
LET'S DERIVE THIS:

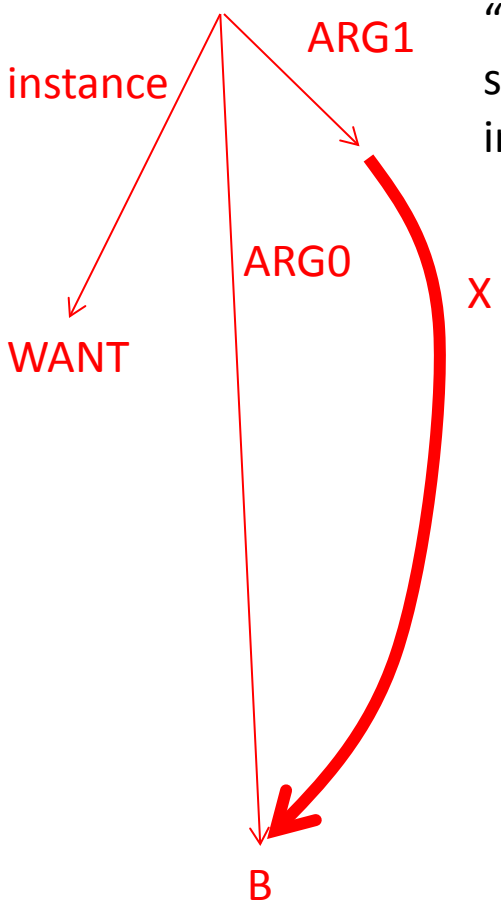


= boy wants girl to believe that he is wanted

HRG Derivation 1

LET'S DERIVE THIS:

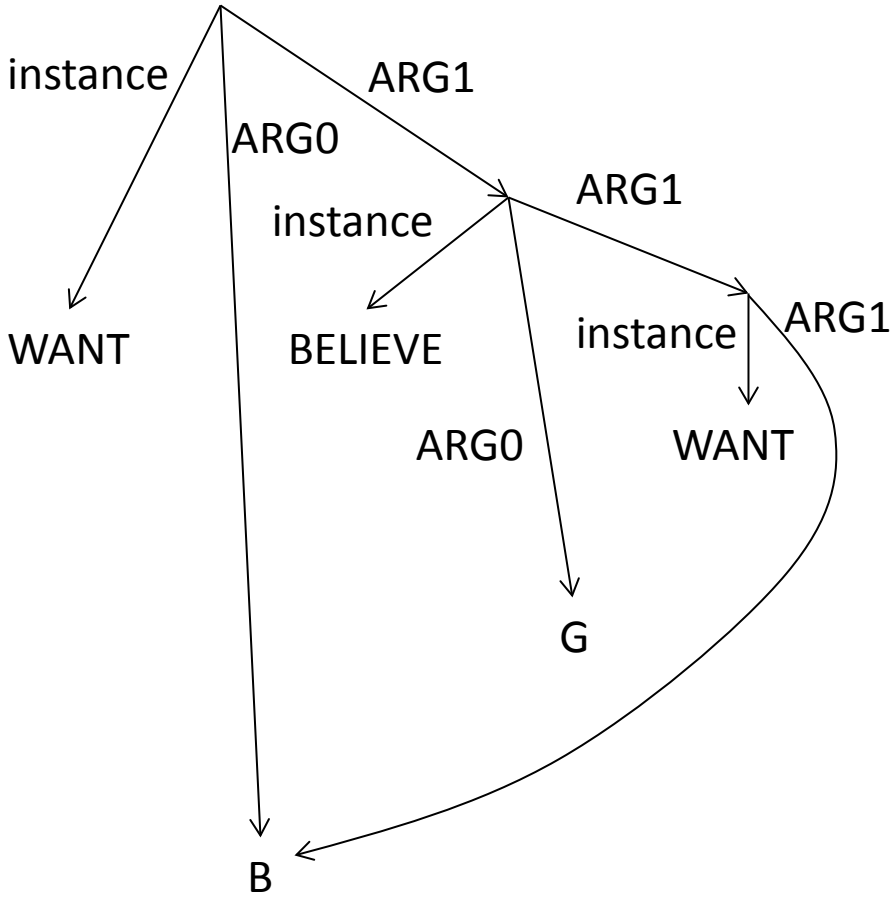


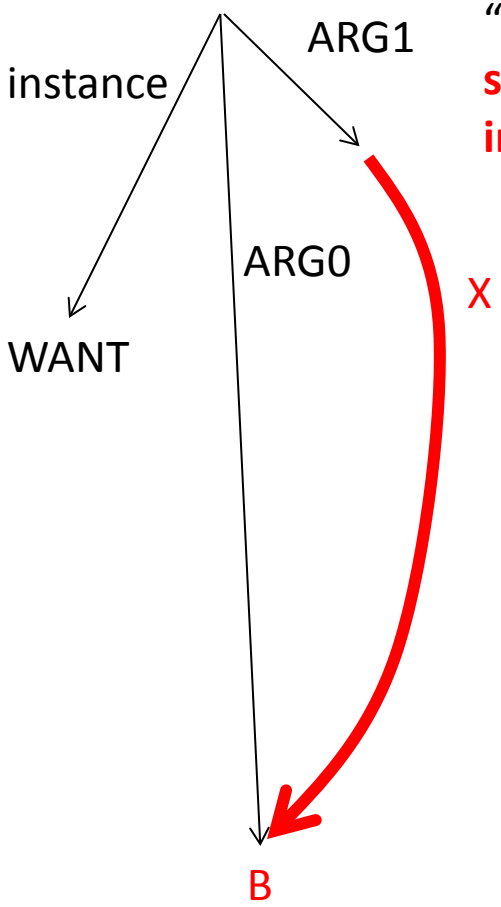


"the boy wants something involving himself"

HRG Derivation 1

LET'S DERIVE THIS:

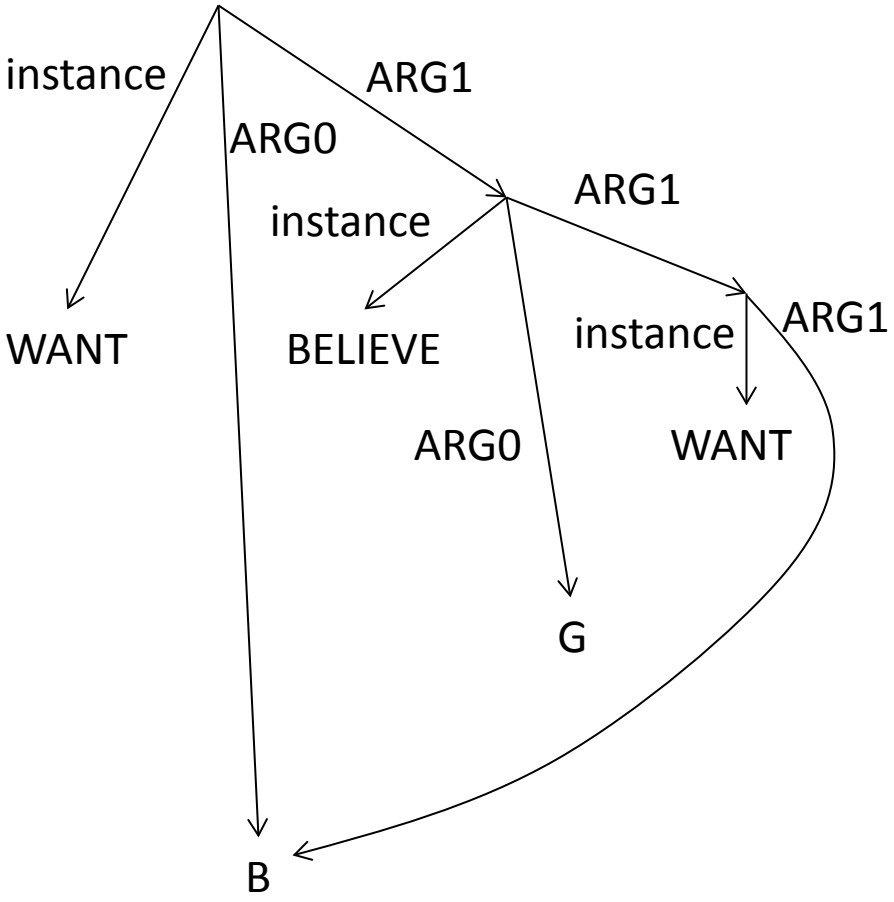




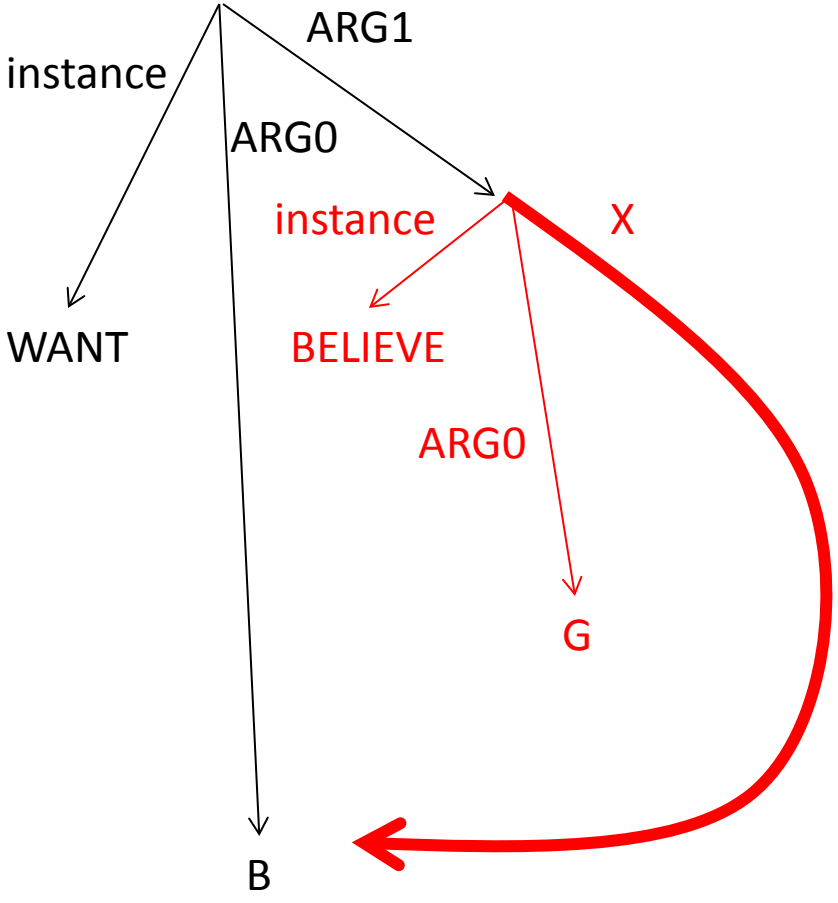
"the boy wants **something involving himself**"

HRG Derivation 1

LET'S DERIVE THIS:

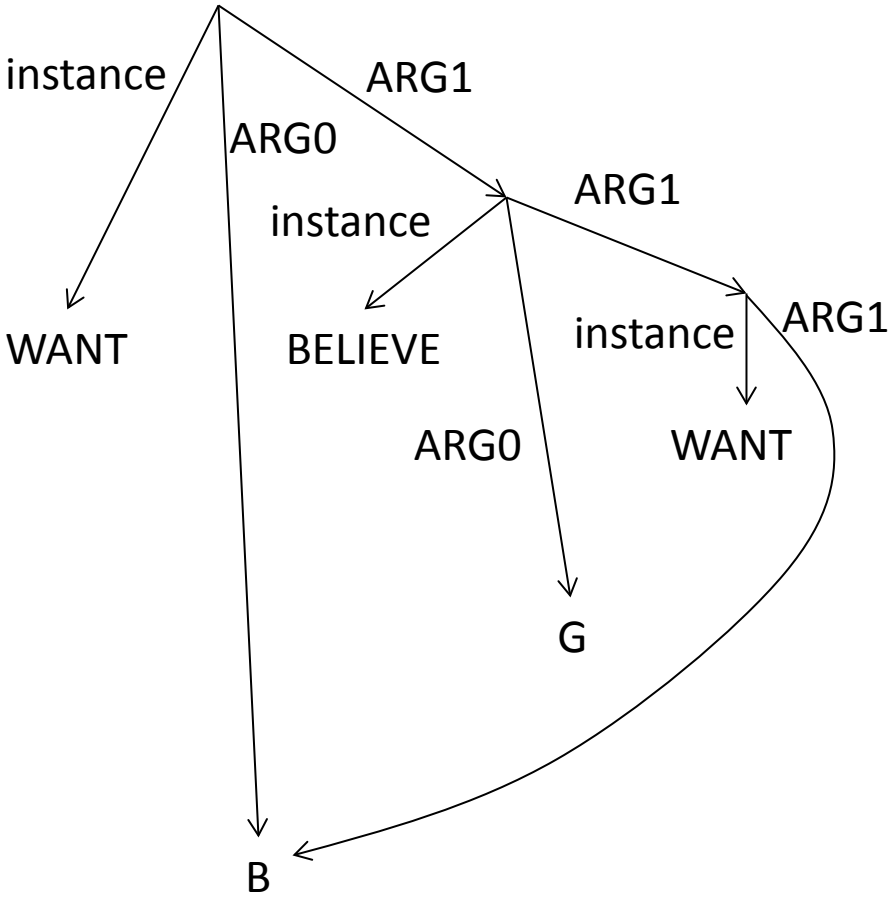


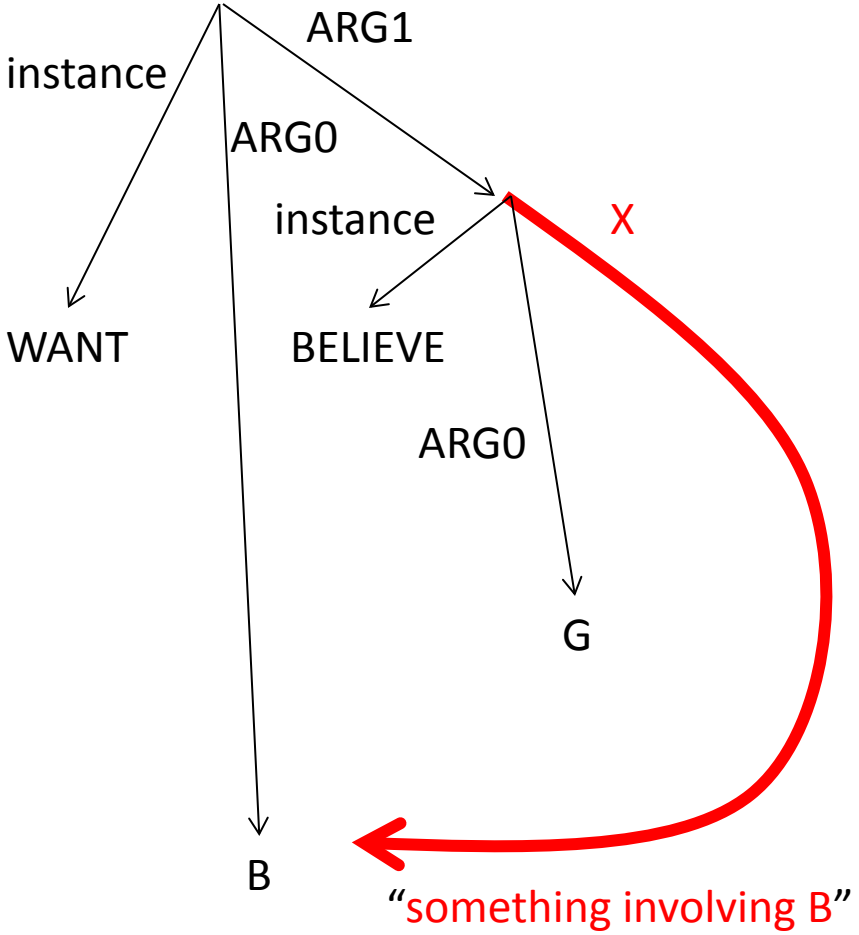
"the boy wants the girl to believe **something involving him**"



HRG Derivation 1

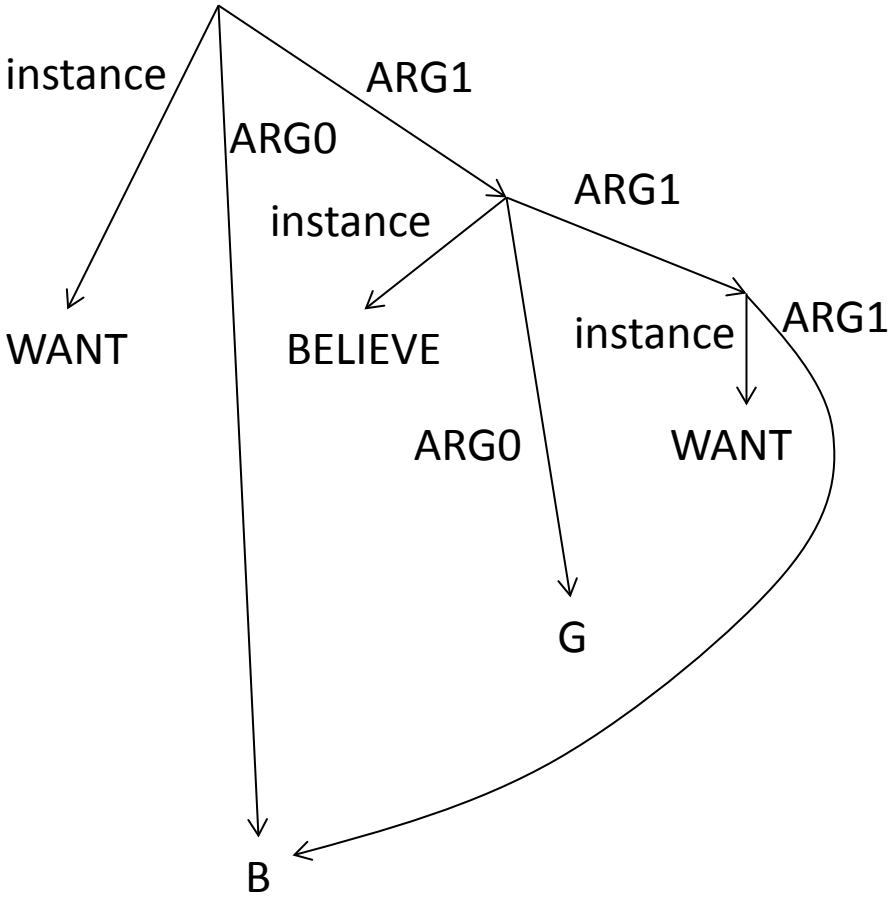
LET'S DERIVE THIS:

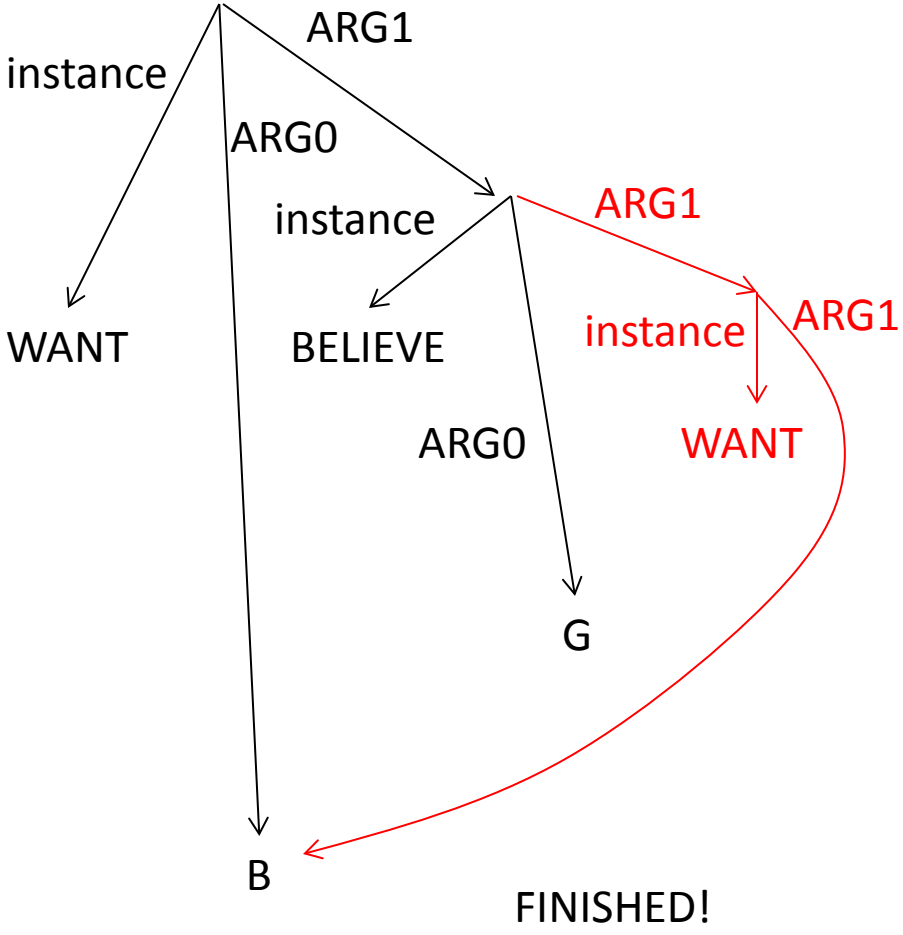




HRG Derivation 1

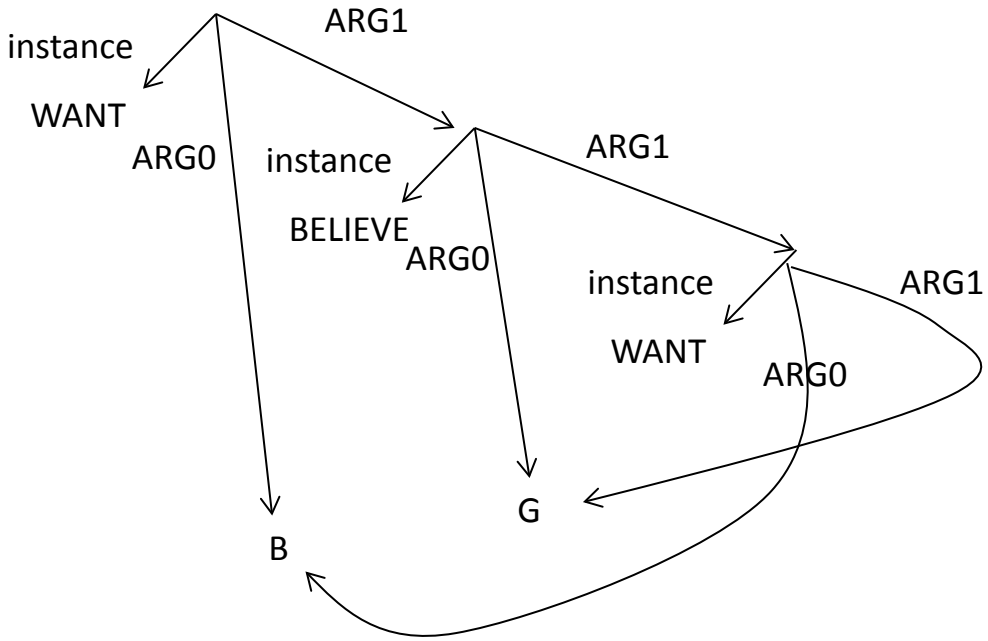
LET'S DERIVE THIS:

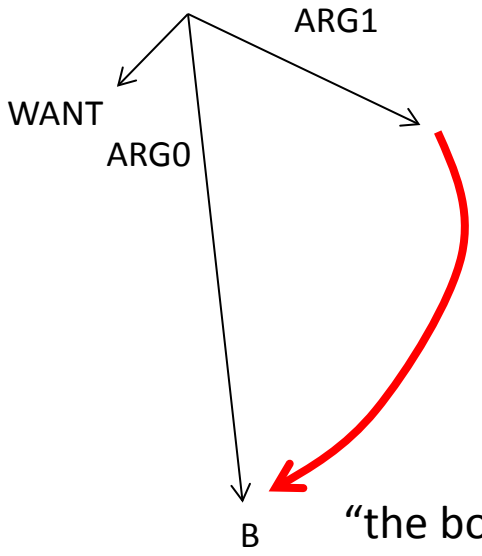




HRG Derivation 2

LET'S DERIVE THIS:



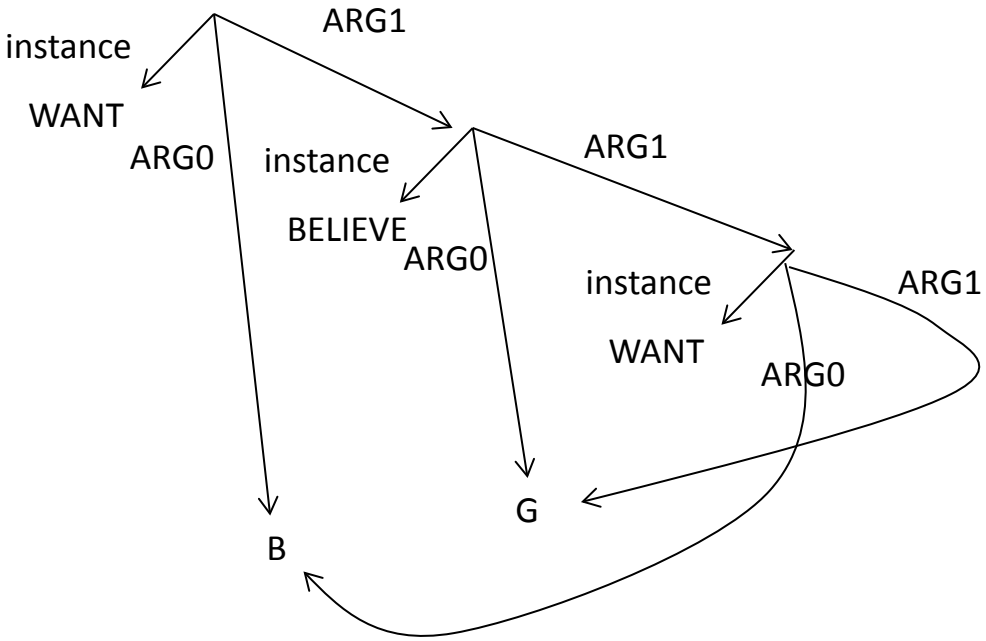


"the boy wants **something involving himself**"

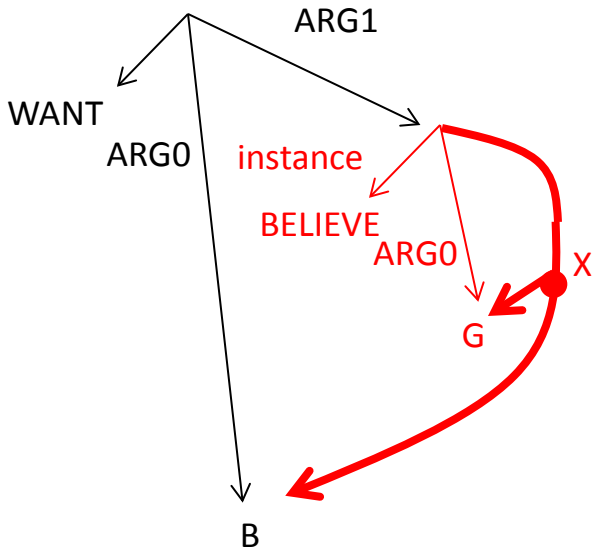
(= boy wants girl to believe that he wants her)

HRG Derivation 2

LET'S DERIVE THIS:



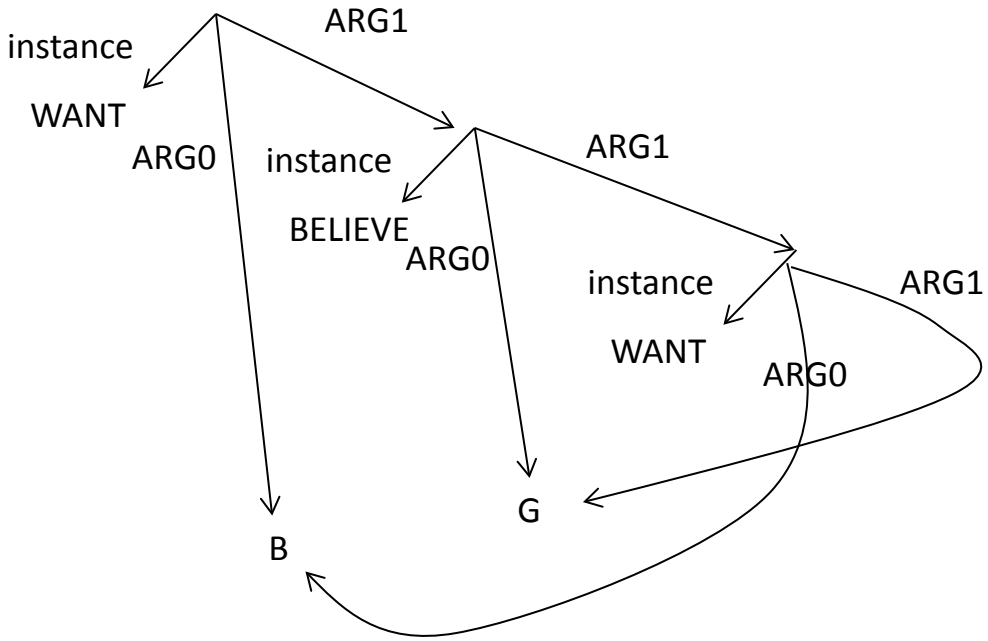
this new hyperedge
labeled X
has two tails

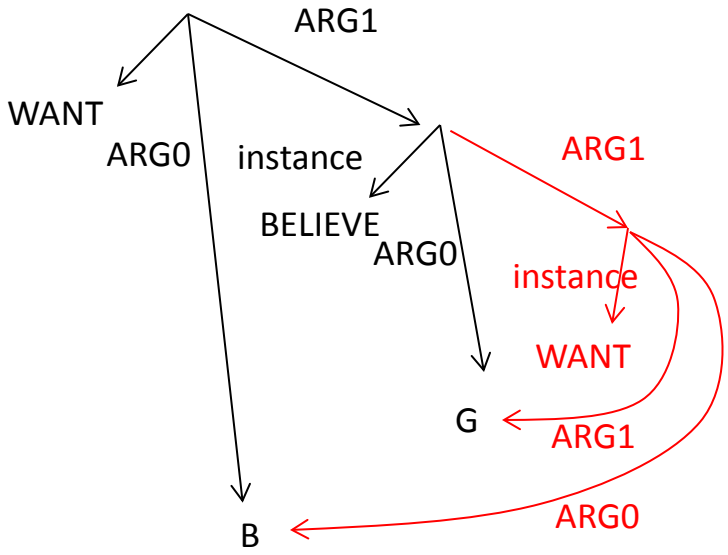


"the boy wants
the girl to believe
**something involving
them both**"

HRG Derivation 2

LET'S DERIVE THIS:





FINISHED!

Formal Properties of HRG Acceptors

		string world		tree world	graph world
		FSA	CFG	RTG	HRG
closure under	union	yes	yes	yes	yes
	intersection	yes	no	yes	no
	complement	yes	no	yes	no
decidability of	emptiness	yes	yes	yes	yes
	equality	yes	no	yes	no

	string world		tree world	graph world
	FSA	CFG	RTG	HRG
membership / EM	$O(n)$	$O(n^3)$	$O(n)$	$O((3^d n)^{r+1})$
k-best derivations	$O(M + klogk)$			

Formal Properties of HRG Acceptors

		string world		tree world	graph world
		FSA	CFG	RTG	HRG
closure under	union	yes	yes	yes	yes
	intersection	yes	no	yes	no
	complement	yes	no	yes	no
decidability of	emptiness	yes	yes	yes	yes
	equality	yes	no	yes	no

mildly unpleasant

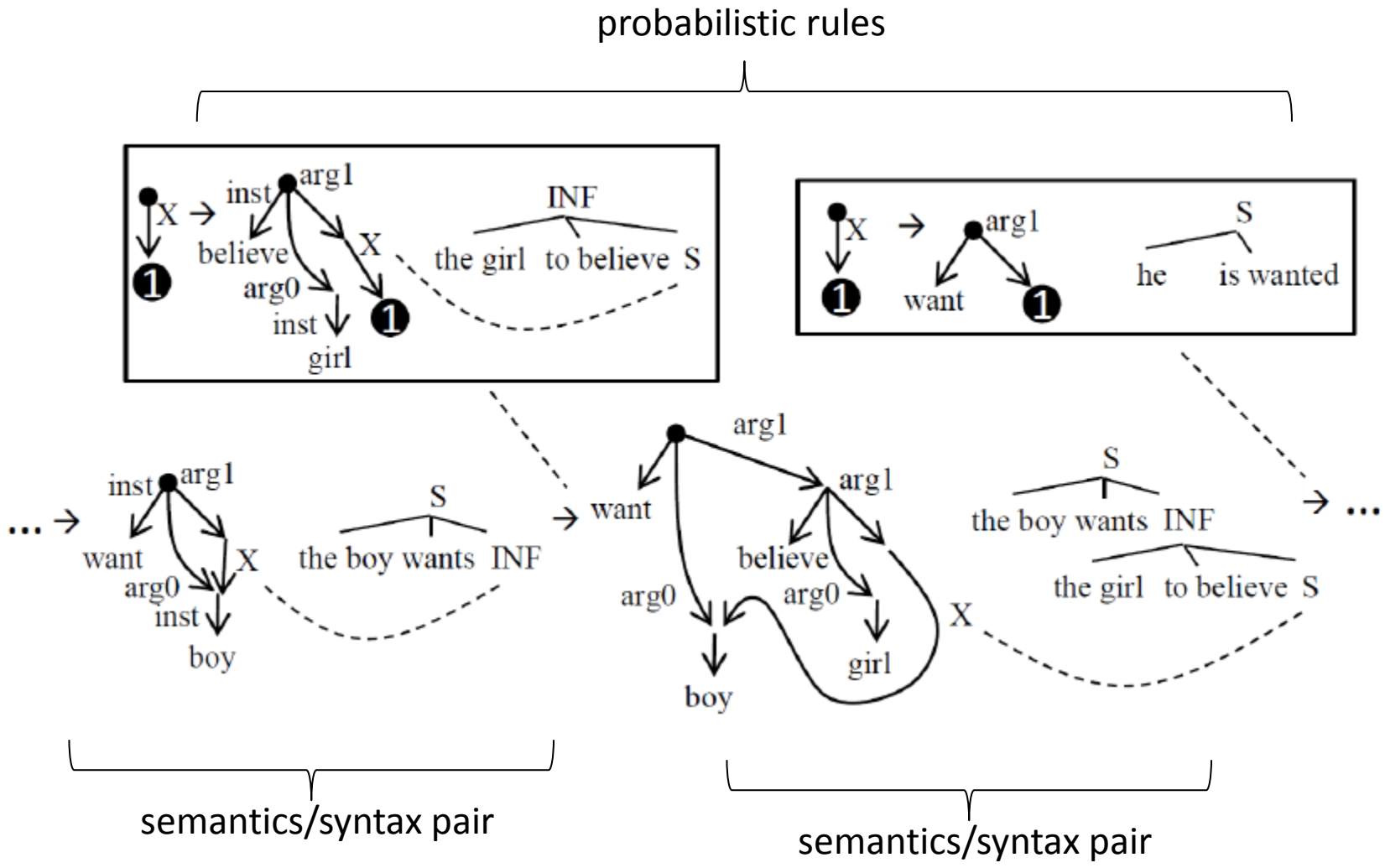
	string world		tree world	graph world
	FSA	CFG	RTG	HRG
membership / EM	$O(n)$	$O(n^3)$	$O(n)$	$O((3^d n)^{T+1})$
k-best derivations	$O(M + klogk)$			

novel algorithm

very pleasant!

d=input graph outdegree
T=treewidth complexity (2-3)

Synchronous HRG



Formal Properties of SHRG Transducers

	string world		tree world		graph world
	FST	SCFG	LNT	xLNT	SHRG
closure under composition	yes	no	yes	no	no
closure under intersection	no	no	no	no	no
transduction preserves recognizability	yes	yes	yes	yes	yes

	string world		tree world	graph world
	FST	SCFG-2	xLNT	SHRG
membership / EM	$O(n^2)$	$O(n^6)$	$O(n^2)$	$O((3^d n)^{T_2+1})$
transduction	$O(n)$	$O(n^3)$	$O(n)$	$O((3^d n)^{T+1})$

Next Things

- Automatic extraction/use of graph grammars
  - Hook up with new manually-created English Sembank of 12,000 sentences (fiction, news, blog)
- Exploration of simpler formal devices
- Graph transduction + NLP workshops (2014):
  - Czech Republic workshop (Chiang, Gildea, Satta)
  - Dagstuhl workshop (Drewes, Knight, Kuhlman)
- AMR to English generation (ISI)
- English to AMR parsing (CMU)

end