

n-Best Parsing Revisited

Matthias BÜchse Daniel Geisler
Torsten Stüber Heiko Vogler

Technische Universität Dresden

Workshop ATANLP
ACL 2010
2010-07-16

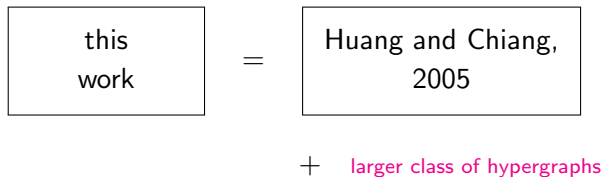
Contribution

this
work

Contribution



Contribution



Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case

Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures

Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures
- + functional specification (Haskell)

Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures
- + functional specification (Haskell)

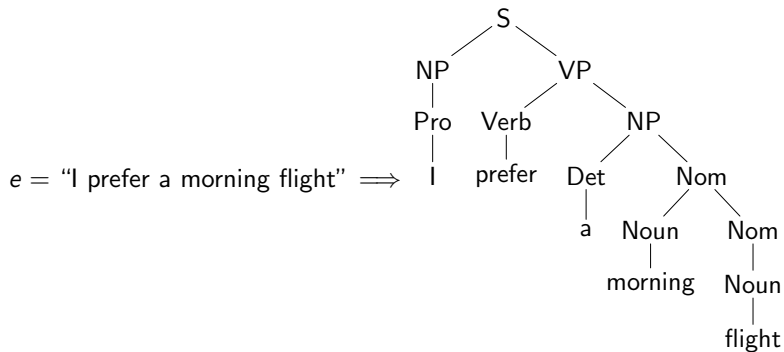
Two Scenarios

- ▶ Parsing
- ▶ Decoding

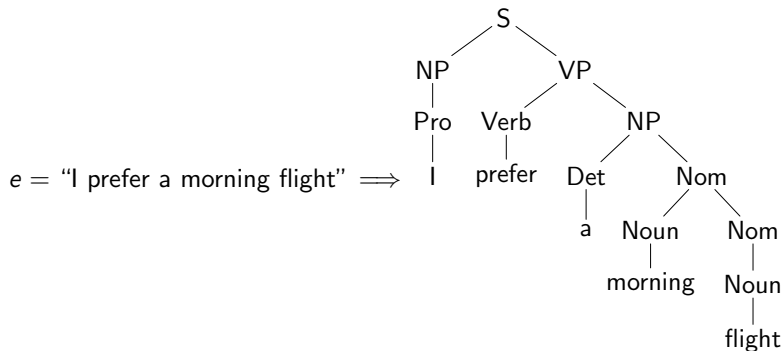
Parsing

$e = \text{"I prefer a morning flight"}$

Parsing



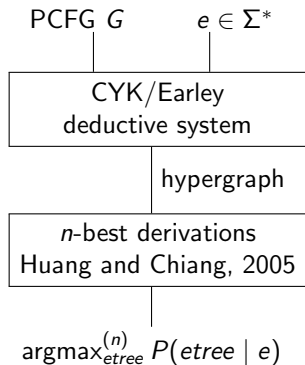
Parsing



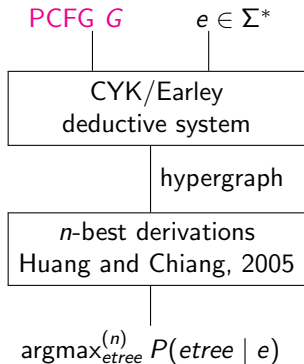
Statistical NLP:

$$\operatorname{argmax}_{\text{etree}} P(\text{etree} \mid e)$$

PCFG Parsing Application

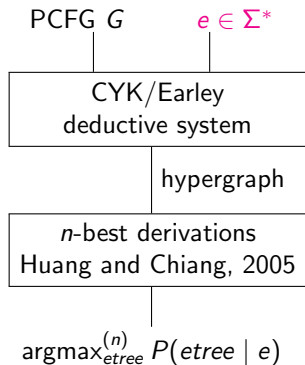


PCFG Parsing Application



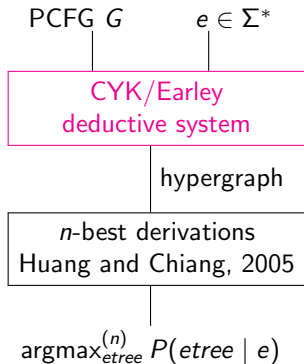
S → NP VP
NP → Pronoun | Proper-Noun | Det Nominal
Nominal → Noun Nominal | Noun
VP → Verb | Verb NP | ...
PP → Preposition NP
Noun → flight | breeze | trip | morning | ...
Verb → is | prefer | like | need | want | fly
Pronoun → me | I | you | it | ...
Proper-Noun → Alaska | Baltimore | ...
Det → the | a | an | this | these | that | ...
Preposition → from | to | on | near | ...
(after Jurafsky and Martin (2000))

PCFG Parsing Application



I prefer a morning flight

PCFG Parsing Application



Item form:
 $[i, A \rightarrow \alpha \bullet \beta, j]$

Goal:
 $[1, S' \rightarrow S \bullet, n+1]$

Rules:

$$\frac{}{[1, S' \rightarrow \bullet S, 1]}$$

Initialization

$$\frac{[i, A \rightarrow \alpha \bullet w_j \beta, j]}{[i, A \rightarrow \alpha w_j \bullet \beta, j+1]}$$

Scanning

$$\frac{R(B \rightarrow \gamma)}{[j, B \rightarrow \bullet \gamma, j]} [i, A \rightarrow \alpha \bullet B \beta, j]$$

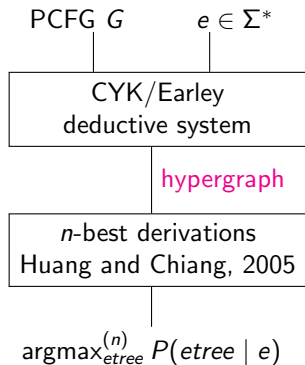
Prediction

$$\frac{[i, A \rightarrow \alpha \bullet B \beta, k] \quad [k, B \rightarrow \gamma \bullet, j]}{[i, A \rightarrow \alpha B \bullet \beta, j]}$$

Completion

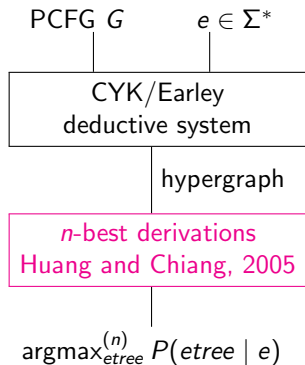
(Goodman, 1999)

PCFG Parsing Application

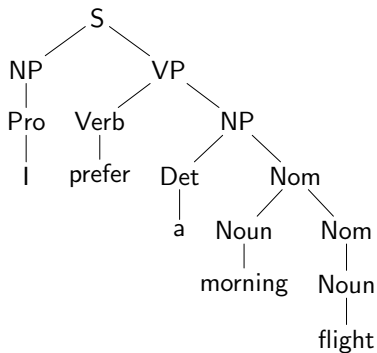
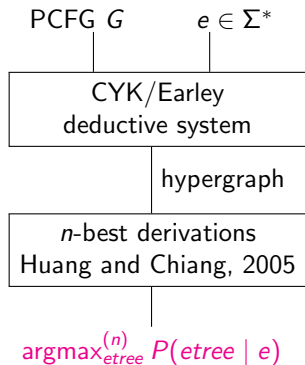


"=" instantiation of the deductive system

PCFG Parsing Application



PCFG Parsing Application



-LM Decoding

$f = \text{"Jean aime Marie"}$

-LM Decoding

$f = \text{"Jean aime Marie"} \implies \text{John loves Mary}$

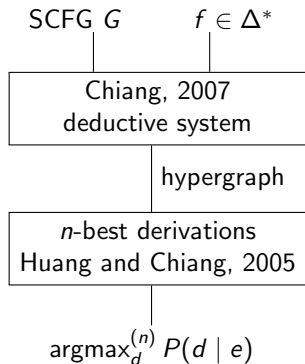
-LM Decoding

$f = \text{"Jean aime Marie"} \implies \text{John loves Mary}$

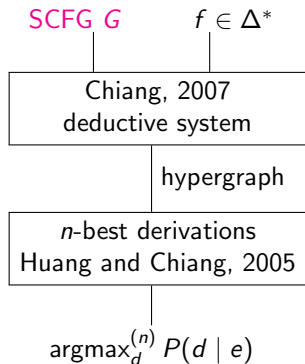
Statistical NLP:

$$\operatorname{argmax}_e P(e | f)$$

SCFG Decoding Application

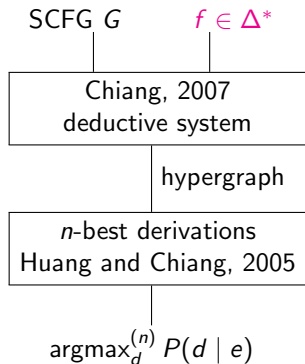


SCFG Decoding Application



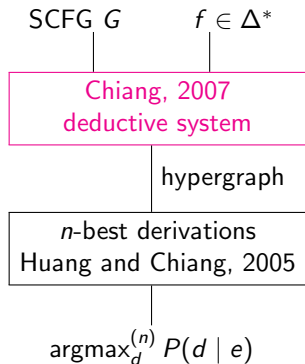
$S \rightarrow \langle S_{[1]} X_{[2]}, S_{[1]} X_{[2]} \rangle$
 $S \rightarrow \langle X_{[1]}, X_{[1]} \rangle$
 $X \rightarrow \langle X_{[1]} X_{[2]}, X_{[2]} X_{[1]} \rangle$
 $X \rightarrow \langle \text{John}, \text{Jean} \rangle$
 $X \rightarrow \langle \text{loves}, \text{aime} \rangle$
 $X \rightarrow \langle \text{Mary}, \text{Marie} \rangle$

SCFG Decoding Application



Jean aime Marie

SCFG Decoding Application



$$\frac{}{X \rightarrow \gamma : w} \quad (X \xrightarrow{w} \langle \gamma, a \rangle) \in G$$

$$\frac{X \rightarrow f_{i+1}^j : w}{[X, i, j] : w}$$

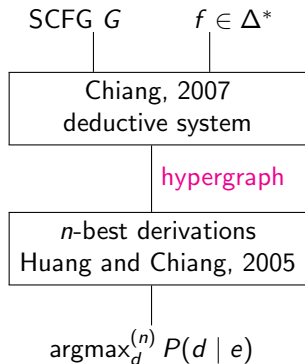
$$\frac{Z \rightarrow f_{i+1}^{i_1} X f_{j_1+1}^j : w \quad [X, i_1, j_1] : w_1}{[Z, i, j] : w w_1}$$

$$\frac{Z \rightarrow f_{i+1}^{i_1} X f_{j_1+1}^{i_2} Y f_{j_2+1}^j : w \quad [X, i_1, j_1] : w_1 \quad [Y, i_2, j_2] : w_2}{[Z, i, j] : w w_1 w_2}$$

Goal item: $[S, 0, n]$

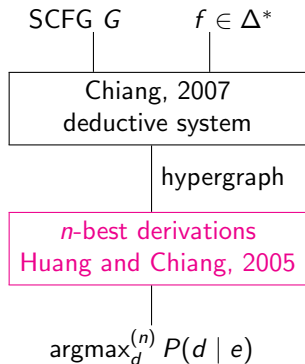
(Chiang, 2007)

SCFG Decoding Application

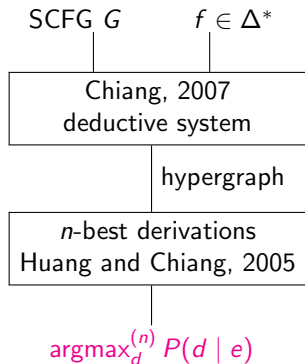


"=" instantiation of the deductive system

SCFG Decoding Application

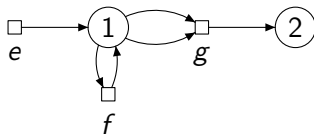


SCFG Decoding Application



John loves Mary

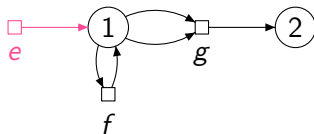
Interlude: Hypergraphs and Derivations



derivations for node 1

derivations for node 2

Interlude: Hypergraphs and Derivations

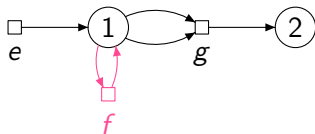


derivations for node 1

derivations for node 2

e

Interlude: Hypergraphs and Derivations

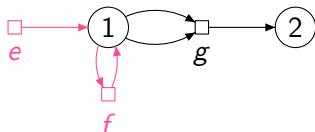


derivations for node 1

derivations for node 2

e
 $f(\dots)$

Interlude: Hypergraphs and Derivations

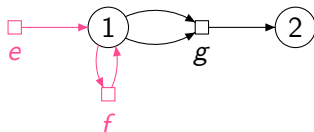


derivations for node 1

derivations for node 2

e
 $f(e)$

Interlude: Hypergraphs and Derivations

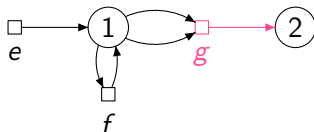


derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

Interlude: Hypergraphs and Derivations



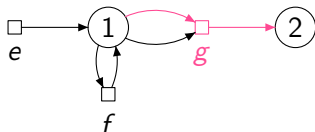
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(\dots, \dots)$

Interlude: Hypergraphs and Derivations



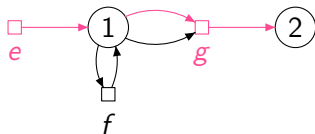
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(\dots, \dots)$

Interlude: Hypergraphs and Derivations



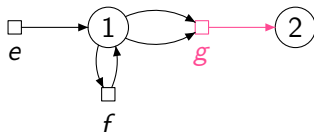
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(e, \dots)$

Interlude: Hypergraphs and Derivations



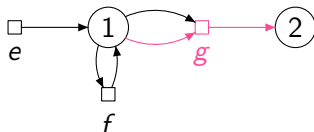
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(e, \dots)$

Interlude: Hypergraphs and Derivations



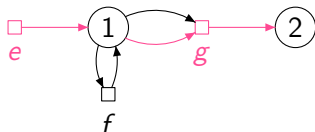
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(e, \dots)$

Interlude: Hypergraphs and Derivations



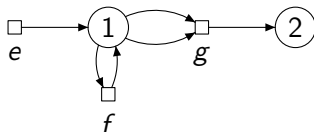
derivations for node 1

derivations for node 2

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

$g(e, e)$

Interlude: Hypergraphs and Derivations



derivations for node 1

e
 $f(e)$
 $f(f(e))$
 $f(f(f(e)))$
 \vdots

derivations for node 2

$g(e, e)$
 $g(e, f(e))$
 $g(f(e), f(e))$
 $g(f(e), e)$
 \vdots

Contribution

this
work

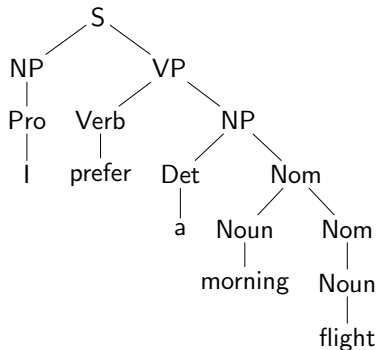
=

Huang and Chiang,
2005

+ larger class of hypergraphs

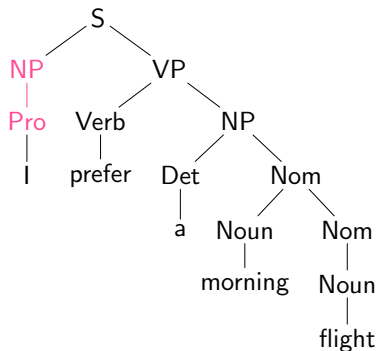
PCFG Parsing Revisited

Subject vs Object NPs



PCFG Parsing Revisited

Subject vs Object NPs

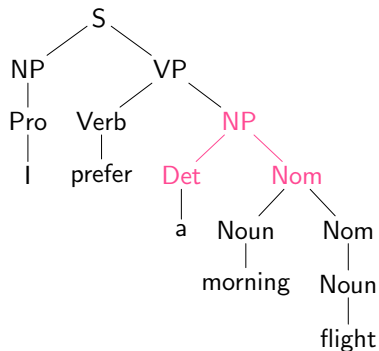


	subject
pronoun	91 %
non-pronoun	9 %

(Jurafsky and Martin, 2009)

PCFG Parsing Revisited

Subject vs Object NPs

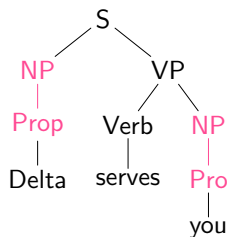


	subject	object
pronoun	91 %	34 %
non-pronoun	9 %	66 %

(Jurafsky and Martin, 2009)

PCFG Parsing Revisited

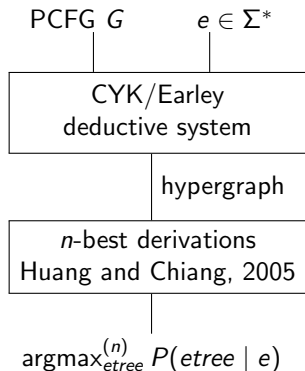
Subject vs Object NPs



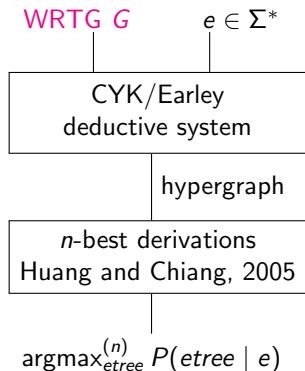
	subject	object
pronoun	91 %	34 %
non-pronoun	9 %	66 %

(Jurafsky and Martin, 2009)

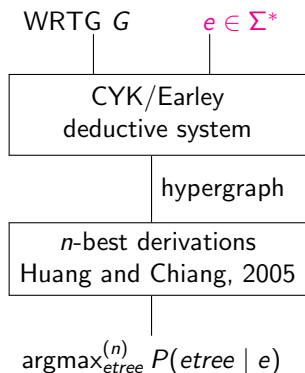
From PCFG to WRTG



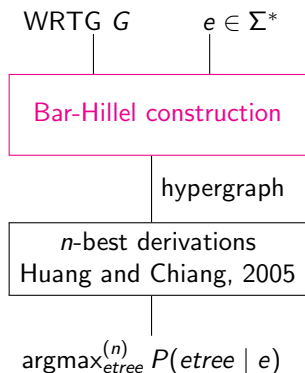
From PCFG to WRTG



From PCFG to WRTG

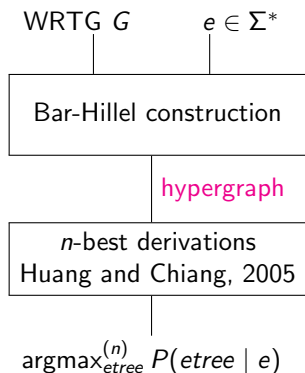


From PCFG to WRTG



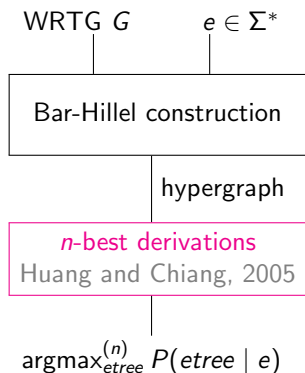
(Maletti and Satta, 2009)

From PCFG to WRTG

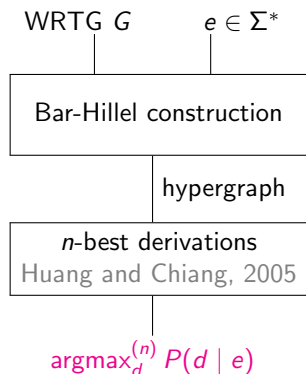


“=” G , but restricted to parse trees of e
multi-edges!

From PCFG to WRTG



From PCFG to WRTG



Contribution

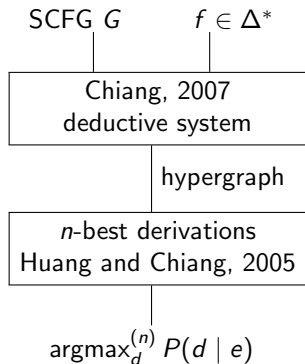
this
work

=

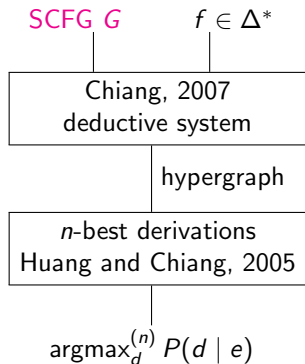
Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case

SCFG Decoding Revisited



SCFG Decoding Revisited



$S \rightarrow \langle S_{[1]} X_{[2]}, S_{[1]} X_{[2]} \rangle$

$S \rightarrow \langle X_{[1]}, X_{[1]} \rangle$

$X \rightarrow \langle \text{super } X_{[1]}, X_{[1]} \rangle$

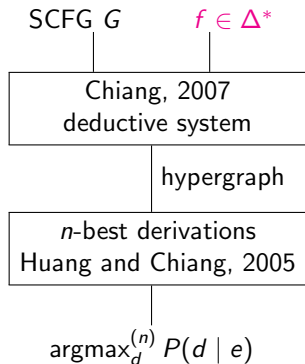
$X \rightarrow \langle X_{[1]} X_{[2]}, X_{[2]} X_{[1]} \rangle$

$X \rightarrow \langle \text{John}, \text{Jean} \rangle$

$X \rightarrow \langle \text{loves}, \text{aime} \rangle$

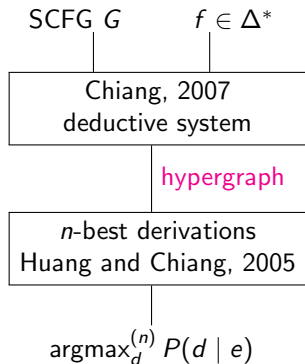
$X \rightarrow \langle \text{Mary}, \text{Marie} \rangle$

SCFG Decoding Revisited



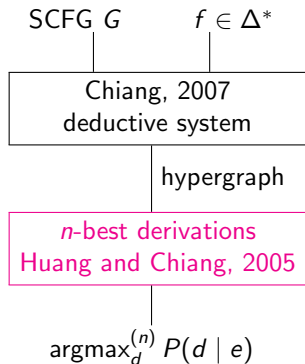
Jean aime Marie

SCFG Decoding Revisited



cyclic!

SCFG Decoding Revisited



conjecture: works in the cyclic case

Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures

Ranking Derivations

1. define interpretation mapping $(.)^I$:

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I()))$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I())) = 1.3$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I())) = 1.3$$

3. define binary relation \succsim , better-than:

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I())) = 1.3$$

3. define binary relation \succsim , better-than:

$$\text{e. g.} \quad \xi_1 \succsim \xi_2 \quad \text{iff} \quad \xi_1^I \leq \xi_2^I$$

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I())) = 1.3$$

3. define binary relation \succsim , better-than:

$$\text{e. g.} \quad \xi_1 \succsim \xi_2 \quad \text{iff} \quad \xi_1^I \leq \xi_2^I$$

Integrate a language model!

Ranking Derivations

1. define interpretation mapping $(.)^I$:

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

2. extend to derivations:

$$[g(e, f(e))]^I = g^I(e^I(), f^I(e^I())) = 1.3$$

3. define binary relation \succsim , better-than:

$$\text{e. g.} \quad \xi_1 \succsim \xi_2 \quad \text{iff} \quad \xi_1^I \leq \xi_2^I$$

\Rightarrow “linear pre-order”

Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

e

$.3$

Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$



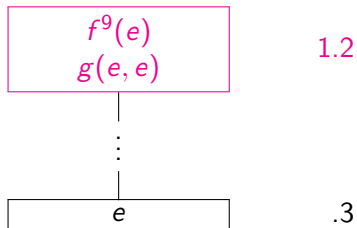
.3

Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

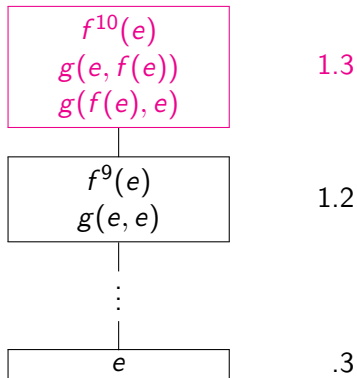


Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

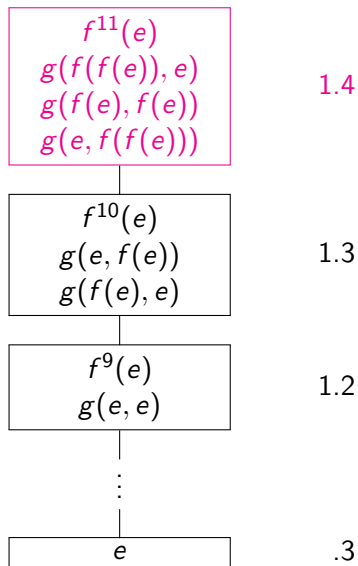


Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$

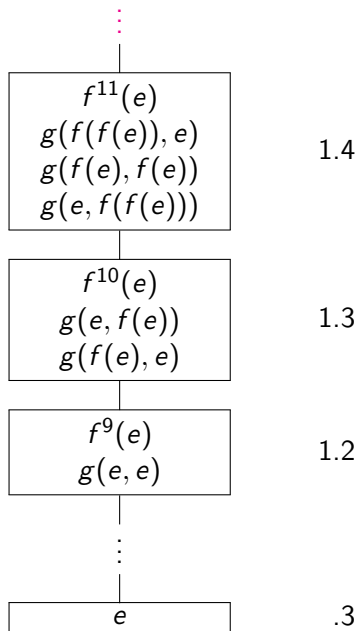


Linear pre-order

$$e^I() = .3$$

$$f^I(x) = x + .1$$

$$g^I(x_1, x_2) = x_1 + x_2 + .6$$



Additional Requirements on the Pre-order

- ▶ Compatibility:

$$\forall i: \xi_i \preceq \xi'_i \implies \sigma(\xi_1, \dots, \xi_k) \preceq \sigma(\xi'_1, \dots, \xi'_k)$$

Additional Requirements on the Pre-order

- ▶ Compatibility:

$$\forall i: \xi_i \preceq \xi'_i \implies \sigma(\xi_1, \dots, \xi_k) \preceq \sigma(\xi'_1, \dots, \xi'_k)$$

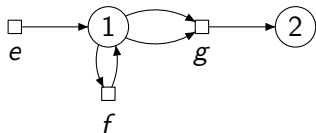
- ▶ Subtree property:

$$\forall i: \xi_i \preceq \sigma(\xi_1, \dots, \xi_k)$$

The n-Best Derivations Problem

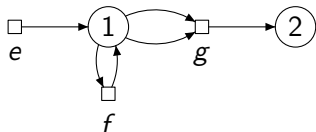
Given

The n-Best Derivations Problem

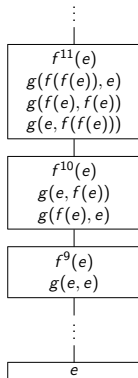


Given a hypergraph H ,

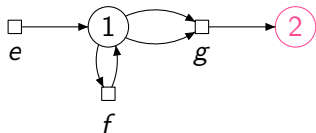
The n-Best Derivations Problem



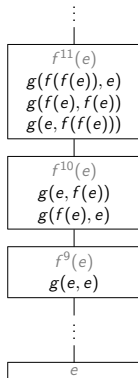
Given a hypergraph H ,
a linear pre-order \succsim
(fulfilling our requirements),



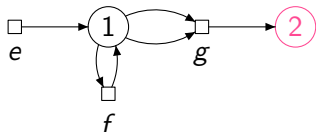
The n-Best Derivations Problem



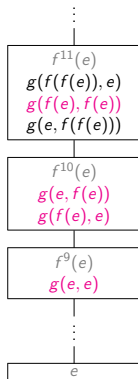
Given a hypergraph H ,
a linear pre-order \preceq
(fulfilling our requirements), and
a target vertex v ,



The n-Best Derivations Problem



Given a hypergraph H ,
a linear pre-order \preceq
(fulfilling our requirements), and
a target vertex v ,
compute n best derivations for v (in order).



Contribution

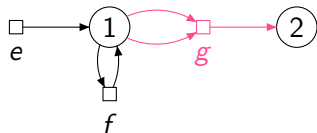
this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures
- + functional specification (Haskell)

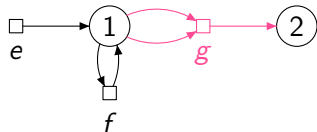
The Top Concatenation Operation



$f(f(f(e)))$				
$f(f(e))$				
$f(e)$				
e				
	e	$f(e)$	$f(f(e))$	$f(f(f(e)))$

$$g^{tc}(l_1, l_2) = g(\text{head } l_1, \text{head } l_2): \\ \text{merge } \{g^{tc}(\text{tail } l_1, [\text{head } l_2]), g^{tc}(l_1, \text{tail } l_2)\}$$

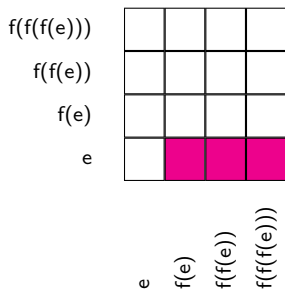
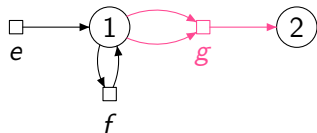
The Top Concatenation Operation



$f(f(f(e)))$				
$f(f(e))$				
$f(e)$				
e				
	e	$f(e)$	$f(f(e))$	$f(f(f(e)))$

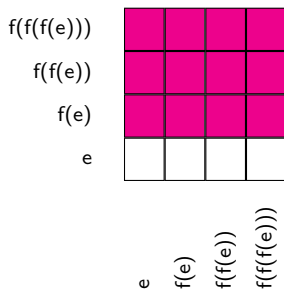
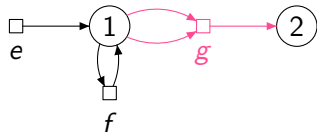
$$g^{tc}(l_1, l_2) = g(\text{head } l_1, \text{head } l_2): \\ \text{merge } \{g^{tc}(\text{tail } l_1, [\text{head } l_2]), g^{tc}(l_1, \text{tail } l_2)\}$$

The Top Concatenation Operation



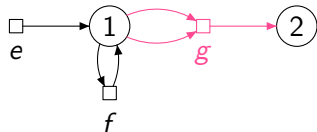
$$g^{tc}(l_1, l_2) = g(\text{head } l_1, \text{head } l_2):$$
$$\text{merge } \{g^{tc}(\text{tail } l_1, [\text{head } l_2]), g^{tc}(l_1, \text{tail } l_2)\}$$

The Top Concatenation Operation



$$g^{tc}(l_1, l_2) = g(\text{head } l_1, \text{head } l_2):$$
$$\text{merge } \{g^{tc}(\text{tail } l_1, [\text{head } l_2]), g^{tc}(l_1, \text{tail } l_2)\}$$

The Top Concatenation Operation



$f(f(f(e)))$				
$f(f(e))$				
$f(e)$				
e				
	e	$f(e)$	$f(f(e))$	$f(f(f(e)))$

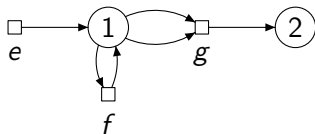
$$g^{\text{tc}}(l_1, l_2) = g(\text{head } l_1, \text{head } l_2):$$

$$\text{merge } \{g^{\text{tc}}(\text{tail } l_1, [\text{head } l_2]), g^{\text{tc}}(l_1, \text{tail } l_2)\}$$

The Merge Operation

$\text{merge } \{l_1, l_2\} =$ if $(\text{head } l_1) \lesssim (\text{head } l_2)$
then $\text{head } l_1 : \text{merge } \{\text{tail } l_1, l_2\}$
else $\text{head } l_2 : \text{merge } \{l_1, \text{tail } l_2\}$

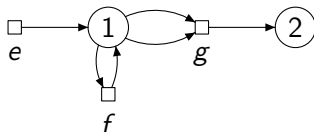
A Provisional Specification



$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$

$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$

A Provisional Specification

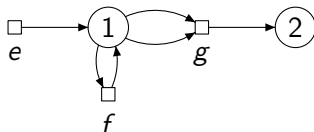


$$p\ 1 = \text{merge} \{e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{\text{tc}}(p\ 1, p\ 1)\}$$

does not work with cyclic hypergraphs

A Provisional Specification



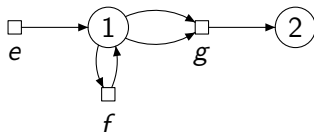
$$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$$

does not work with cyclic hypergraphs

head (p 1)

A Provisional Specification



$$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$$

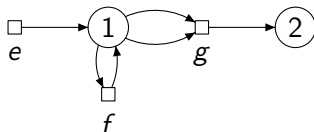
$$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$$

does not work with cyclic hypergraphs

head (p 1)

$$= \text{head} (\text{merge} \{e^{tc}(), f^{tc}(p\ 1)\})$$

A Provisional Specification



$$p\ 1 = \text{merge} \{e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{\text{tc}}(p\ 1, p\ 1)\}$$

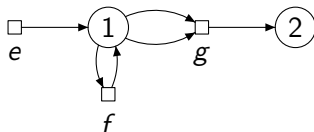
does not work with cyclic hypergraphs

head (p 1)

= head (merge $\{e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$)

= if (head $e^{\text{tc}}()$) \lesssim (head $f^{\text{tc}}(p\ 1)$) ...

A Provisional Specification



$$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$$

does not work with cyclic hypergraphs

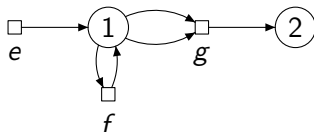
head (p 1)

$$= \text{head} (\text{merge} \{e^{tc}(), f^{tc}(p\ 1)\})$$

$$= \text{if} (\text{head } e^{tc}()) \lesssim (\text{head } f^{tc}(p\ 1)) \dots$$

$$= \text{if } e \lesssim f(\text{head } (p\ 1)) \dots$$

A Provisional Specification



$$p\ 1 = \text{merge} \{e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{\text{tc}}(p\ 1, p\ 1)\}$$

does not work with cyclic hypergraphs

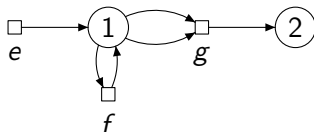
head (p 1)

$$= \text{head} (\text{merge} \{e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\})$$

$$= \text{if} (\text{head } e^{\text{tc}}()) \lesssim (\text{head } f^{\text{tc}}(p\ 1)) \dots$$

$$= \text{if } e \lesssim f(\text{head } (p\ 1)) \dots$$

A Provisional Specification

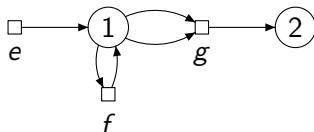


$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$

$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$

works with acyclic hypergraphs

A Provisional Specification



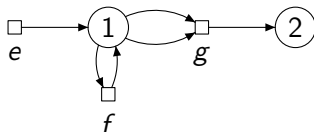
$$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$$

$$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$$

works with acyclic hypergraphs

- ▶ strict evaluation/call-by-value “ \approx ” bottom-up
“ \approx ” Algorithm 2 of (Huang and Chiang, 2005)

A Provisional Specification



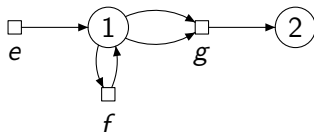
$p\ 1 = \text{merge} \{e^{tc}(), f^{tc}(p\ 1)\}$

$p\ 2 = \text{merge} \{g^{tc}(p\ 1, p\ 1)\}$

works with acyclic hypergraphs

- ▶ strict evaluation/call-by-value “ \approx ” bottom-up
“ \approx ” Algorithm 2 of (Huang and Chiang, 2005)
- ▶ lazy evaluation/call-by-need “ \approx ” top-down
“ \approx ” Algorithm 3 of (Huang and Chiang, 2005)

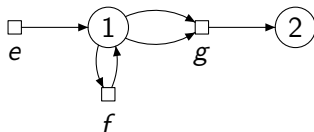
Final Specification



$p\ 1 = (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$

$p\ 2 = (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\}$

Final Specification

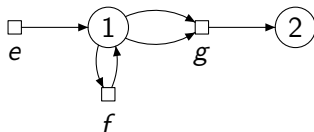


$$p\ 1 = (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$p\ 2 = (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\}$$

- ▶ algorithms for solving the 1-best-derivations problem well established

Final Specification



$$p\ 1 = (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$p\ 2 = (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\}$$

- ▶ algorithms for solving the 1-best-derivations problem well established
- ▶ cf. (Knuth 1977)

Example Computation

p 1 =

p 2 =

Example Computation

$p_1 =$

$p_2 = (\text{head } g^{\text{tc}}(p_1, p_1)):\text{merge } \{\text{tail } g^{\text{tc}}(p_1, p_1)\}$

Example Computation

$p_1 =$

$p_2 = (\text{head } g^{\text{tc}}(p_1, p_1)):\text{merge } \{\text{tail } g^{\text{tc}}(p_1, p_1)\}$

Example Computation

$p\ 1 =$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$p_1 =$

$p_2 = (\text{head } g^{\text{tc}}(p_1, p_1)):\text{merge } \{\text{tail } g^{\text{tc}}(p_1, p_1)\}$
 $= g(\text{head } (p_1), \text{head } (p_1)):\text{merge } \{\text{tail } g^{\text{tc}}(p_1, p_1)\}$

Example Computation

$$p\ 1 = (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$$p\ 1 = (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$$\begin{aligned} p\ 1 &= (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \\ &= e:\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \end{aligned}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$$\begin{aligned} p\ 1 &= (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \\ &= e:\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \end{aligned}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(e, e):\text{head } (\text{tail } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail}^2 g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$$\begin{aligned} p\ 1 &= (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \\ &= e:\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \end{aligned}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(e, e):\text{head } (\text{tail } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail}^2 g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Example Computation

$$\begin{aligned} p\ 1 &= (\text{head } e^{\text{tc}}()):\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \\ &= e:\text{merge } \{\text{tail } e^{\text{tc}}(), f^{\text{tc}}(p\ 1)\} \end{aligned}$$

$$\begin{aligned} p\ 2 &= (\text{head } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(\text{head } (p\ 1), \text{head } (p\ 1)):\text{merge } \{\text{tail } g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(e, e):\text{head } (\text{tail } g^{\text{tc}}(p\ 1, p\ 1)):\text{merge } \{\text{tail}^2 g^{\text{tc}}(p\ 1, p\ 1)\} \\ &= g(e, e):\text{head } (\text{merge } \{g^{\text{tc}}(\text{tail } (p\ 1), [\text{head } (p\ 1)]), g^{\text{tc}}(p\ 1, \text{tail } (p\ 1))\}) \\ &\quad :\text{merge } \{\text{tail}^2 g^{\text{tc}}(p\ 1, p\ 1)\} \end{aligned}$$

Complexity and Runtime

Contribution

this
work

=

Huang and Chiang,
2005

- + larger class of hypergraphs
- + confirm conjecture for cyclic case
- + more flexible weight structures
- + functional specification (Haskell)

References



David Chiang.

2007.

Hierarchical phrase-based translation. *Comp. Ling.*, 33(2):201–228.



Joshua Goodman.

1999.

Semiring parsing. *Comp. Ling.*, 25(4):573–605.



Liang Huang and David Chiang.

2005.

Better k-best parsing. In *Proc. 9th Int. Conf. Parsing Technologies*, pages 53–64. ACL.



Daniel Jurafsky and James H. Martin.

2009.

Speech and Language Processing – [. . .]. Prentice-Hall, 2nd edition.



Andreas Maletti and Giorgio Satta.

2009.

Parsing algorithms based on tree automata. In *Proc. 11th Int. Conf. Parsing Technologies*, pages 1–12. ACL.