Human-Inspired Structured Prediction for Language and Biology

Baidu Research

Liang Huang

Principal Scientist, Baidu Research
Assistant Professor, Oregon State University

Oregon State University

incremental & linear-time

Human-Inspired Structured Prediction for Language and Biology

Liang Huang

Principal Scientist, Baidu Research
Assistant Professor, Oregon State University

incremental & linear-time

Human-Inspired Structured Prediction for Language and Biology

simultaneous interpretation

Liang Huang

Principal Scientist, Baidu Research
Assistant Professor, Oregon State University

Baidu Research

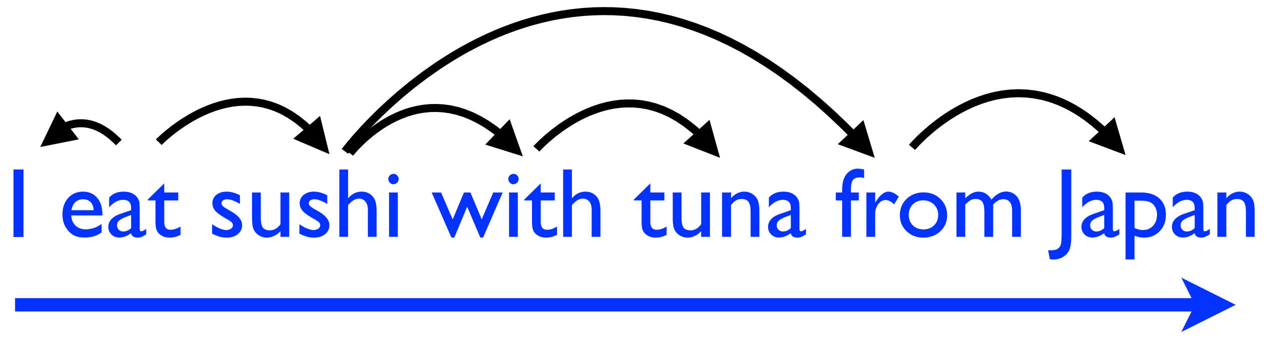
Oregon State University

incremental & linear-time

Human-Inspired Structured Prediction for Language and Biology

simultaneous interpretation

syntactic structure



natural language sequence

Baidu Research

Liang Huang

Principal Scientist, Baidu Research

Assistant Professor, Oregon State University

Oregon State University

incremental & linear-time

Human-Inspired Structured Prediction for Language and Biology

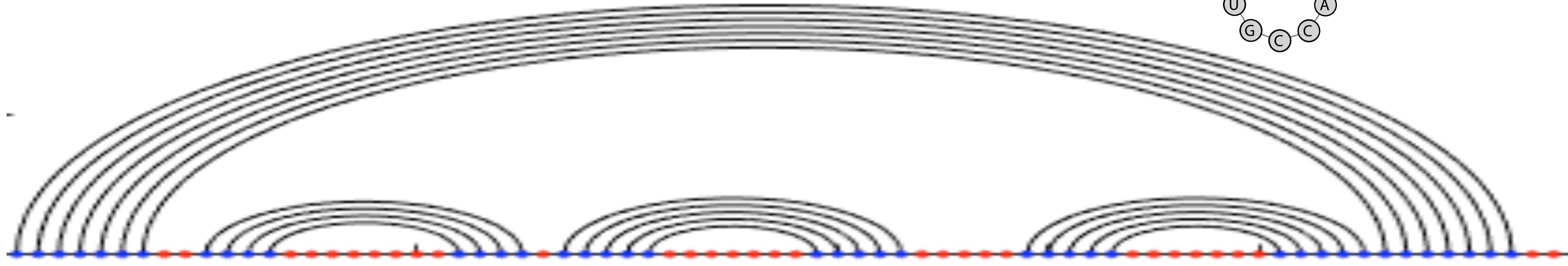
simultaneous interpretation

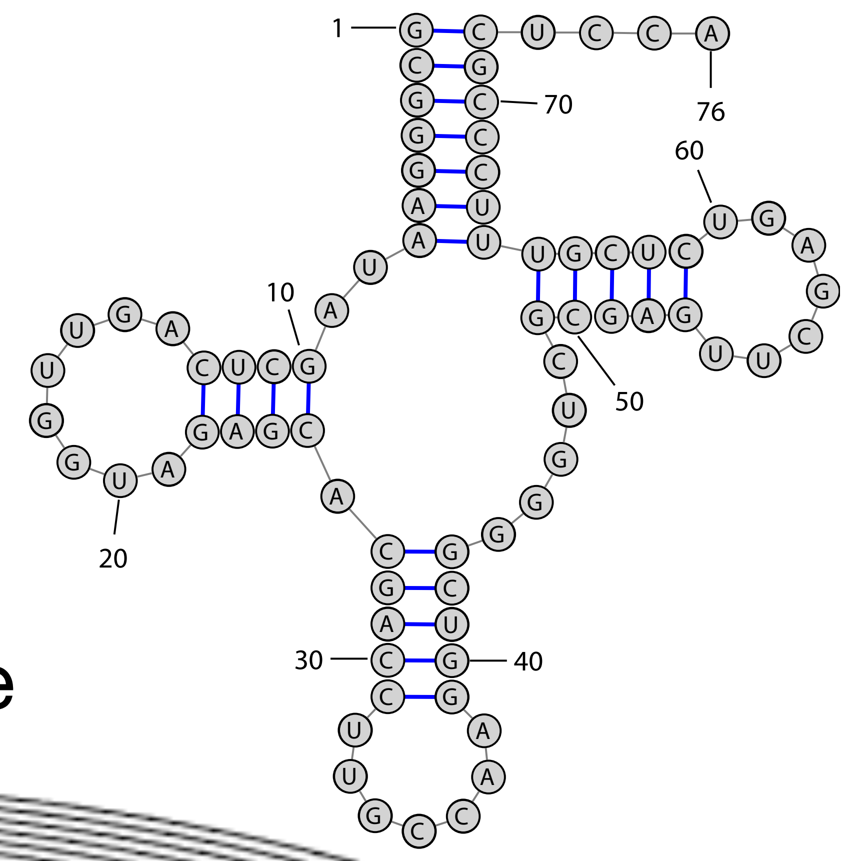
syntactic structure

I eat sushi with tuna from Japan

natural language sequence

secondary structure





GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUC

RNA sequence

Baidu Research

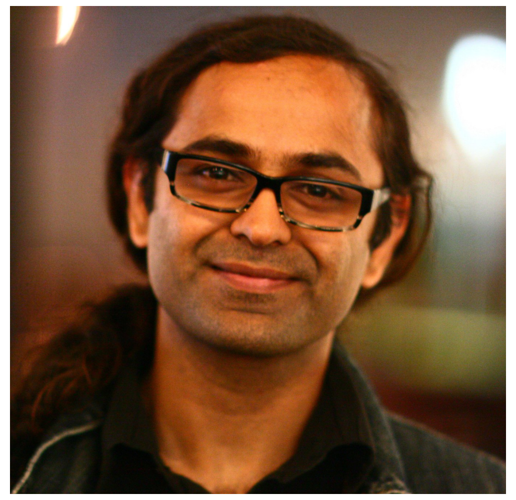
Liang Huang

Principal Scientist, Baidu Research
Assistant Professor, Oregon State University

Oregon State University

A Bit about Myself…

My PhD Graduates



Ashish Vaswani (USC, 2014)
(co-advised by David Chiang)

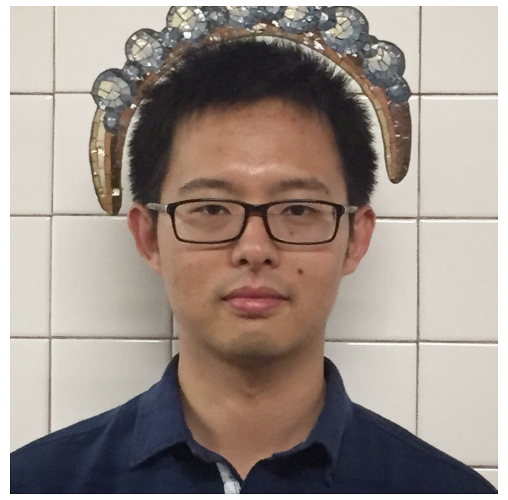
Senior Research Scientist
Google Brain

first author of Transformer
"*Attention is All You Need*"

James Cross (OSU, 2016)

Research Scientist
Facebook

EMNLP 2016 Best Paper
Honorable Mention



Kai Zhao (OSU, 2017)

Research Scientist
Google

11 top-conference papers
(ACL/EMNLP/NAACL)

Mingbo Ma (OSU, 2018)

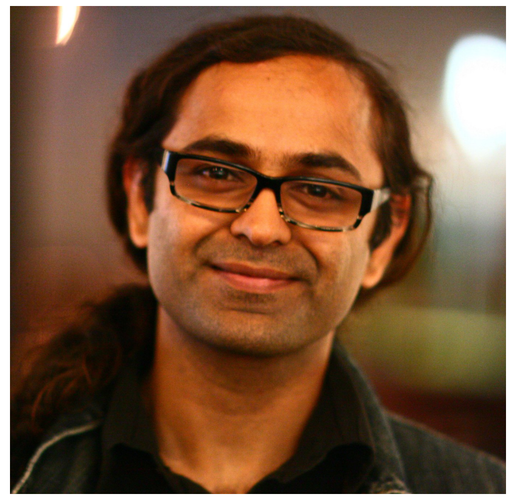
Research Scientist
Baidu Research USA

breakthrough in
simultaneous translation

…

A Bit about Myself…

My PhD Graduates



Ashish Vaswani (USC, 2014)
(co-advised by David Chiang)

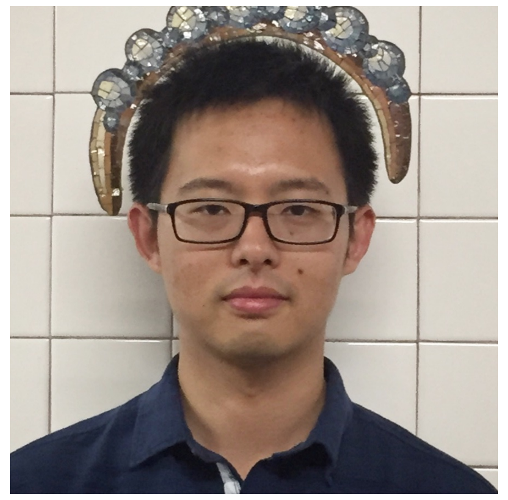
Senior Research Scientist
Google Brain

first author of Transformer
"*Attention is All You Need*"

James Cross (OSU, 2016)

Research Scientist
Facebook

EMNLP 2016 Best Paper
Honorable Mention



Kai Zhao (OSU, 2017)

Research Scientist
Google

11 top-conference papers
(ACL/EMNLP/NAACL)

Mingbo Ma (OSU, 2018)

Research Scientist
Baidu Research USA

breakthrough in
simultaneous translation

…

My Research: Efficient Structured Prediction

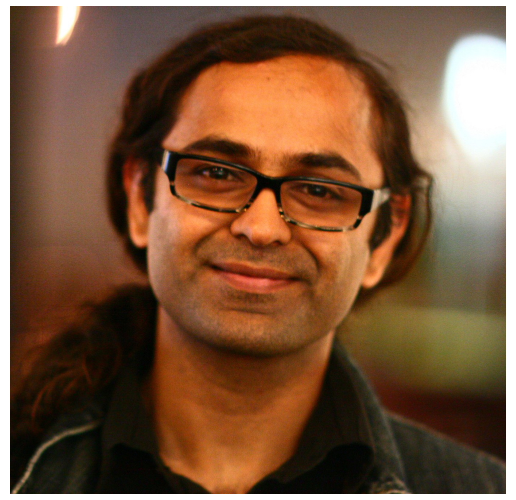
Bush met Putin in Moscow — source language sentence

I eat sushi with tuna from Japan — natural language sentence

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA — RNA sequence

A Bit about Myself…

My PhD Graduates



Ashish Vaswani (USC, 2014)
(co-advised by David Chiang)

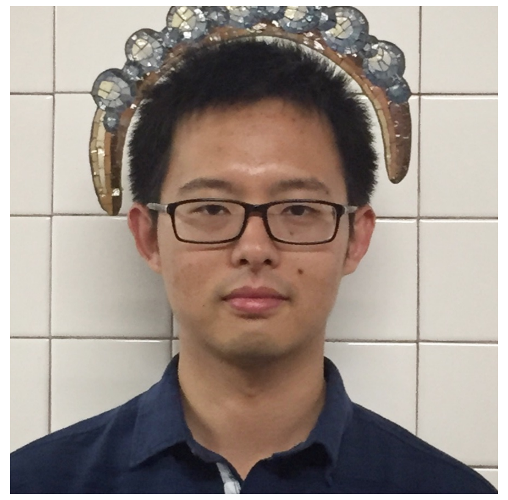
Senior Research Scientist
Google Brain

first author of Transformer
“Attention is All You Need”

James Cross (OSU, 2016)

Research Scientist
Facebook

EMNLP 2016 Best Paper
Honorable Mention



Kai Zhao (OSU, 2017)

Research Scientist
Google

11 top-conference papers
(ACL/EMNLP/NAACL)

Mingbo Ma (OSU, 2018)

Research Scientist
Baidu Research USA

breakthrough in
simultaneous translation

…

My Research: Efficient Structured Prediction

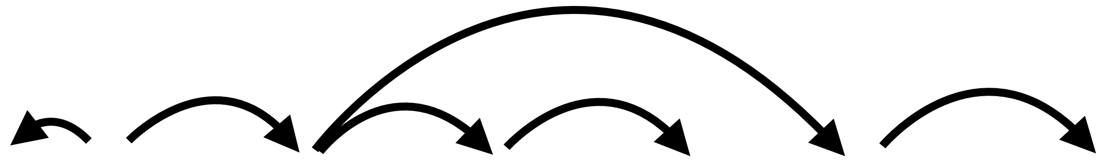
target-language sequence

布什在莫斯科与普京会晤

Bush met Putin in Moscow

source language sentence

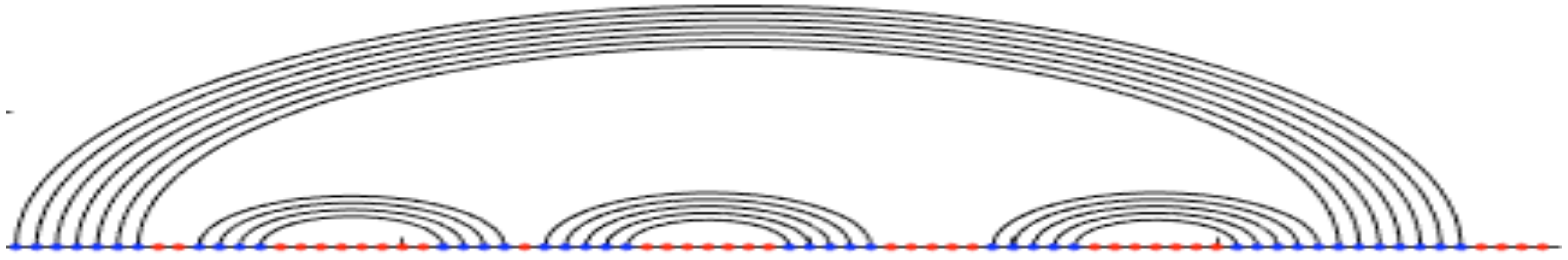
syntactic structure



I eat sushi with tuna from Japan

natural language sentence

secondary structure

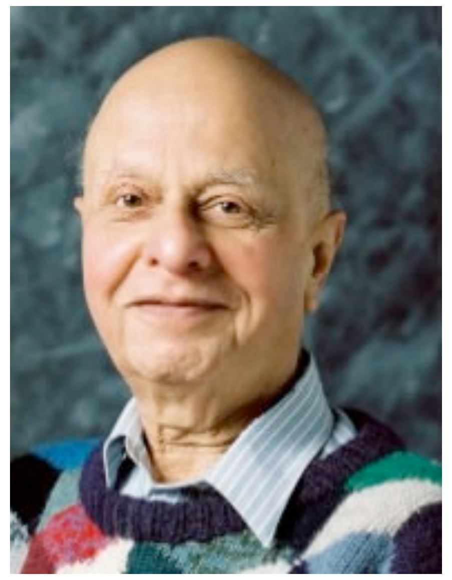


GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

RNA sequence

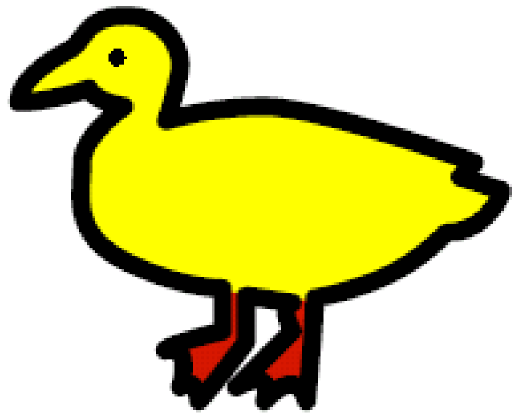
Language is Hard, Even for Humans

- *how many interpretations?*



Aravind Joshi
(1929-2018)

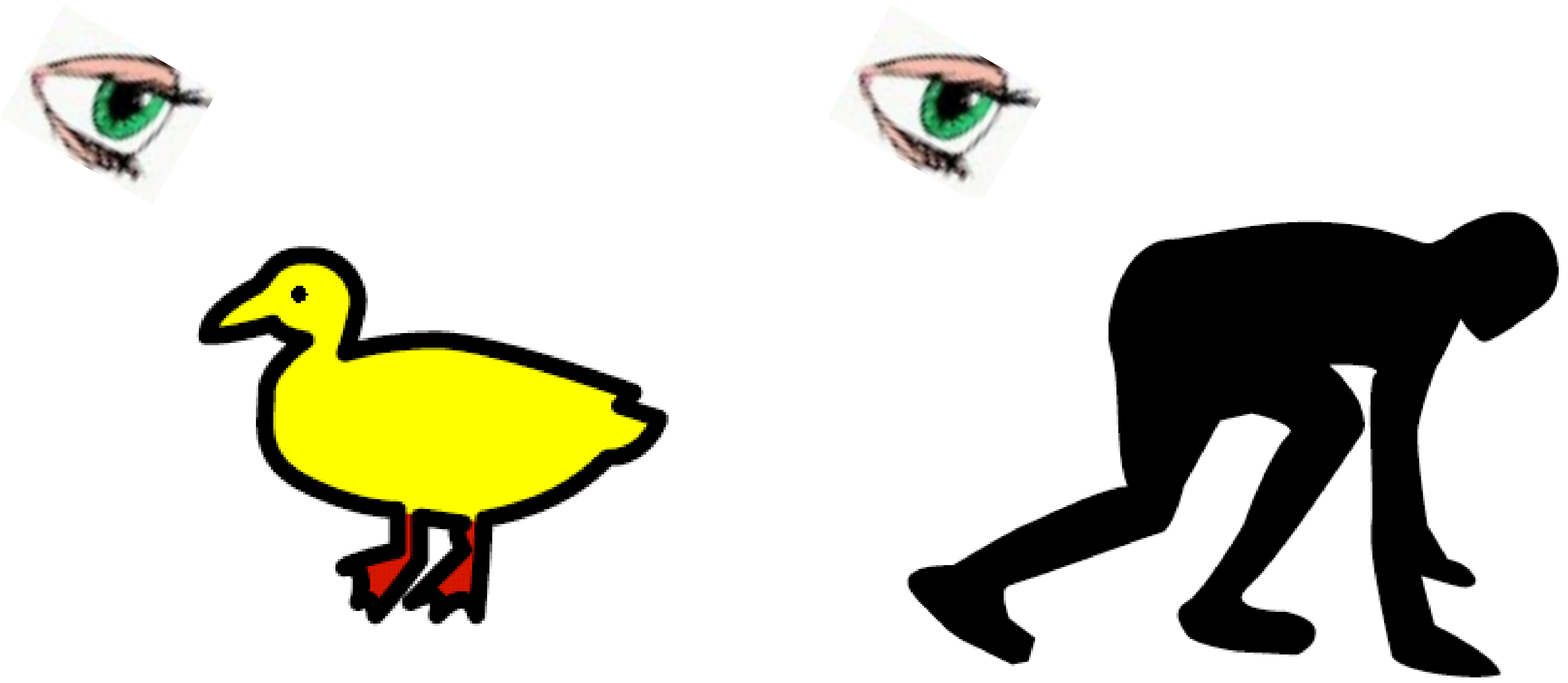
I saw her duck

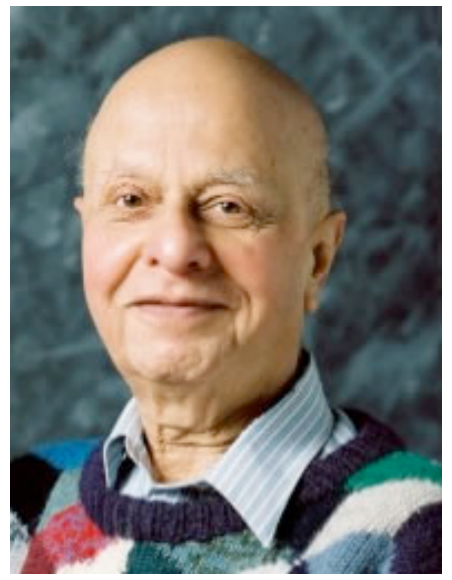


Language is Hard, Even for Humans

- *how many interpretations?*

I saw her duck

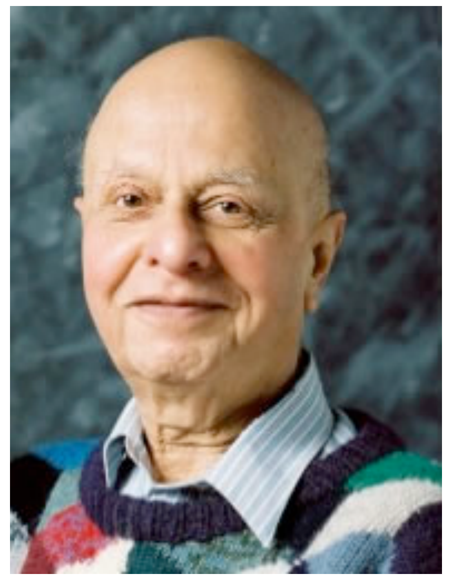




Aravind Joshi
(1929-2018)

Language is Hard, Even for Humans

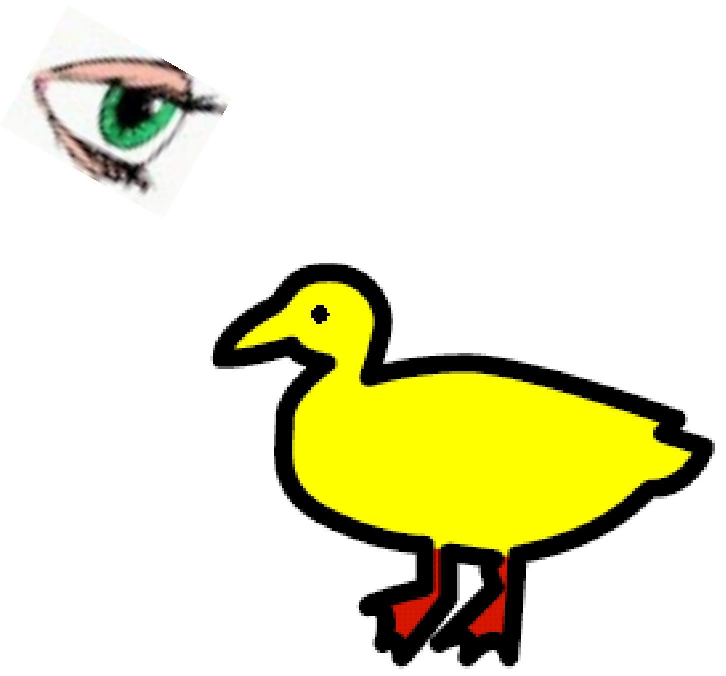
- *how many interpretations?*

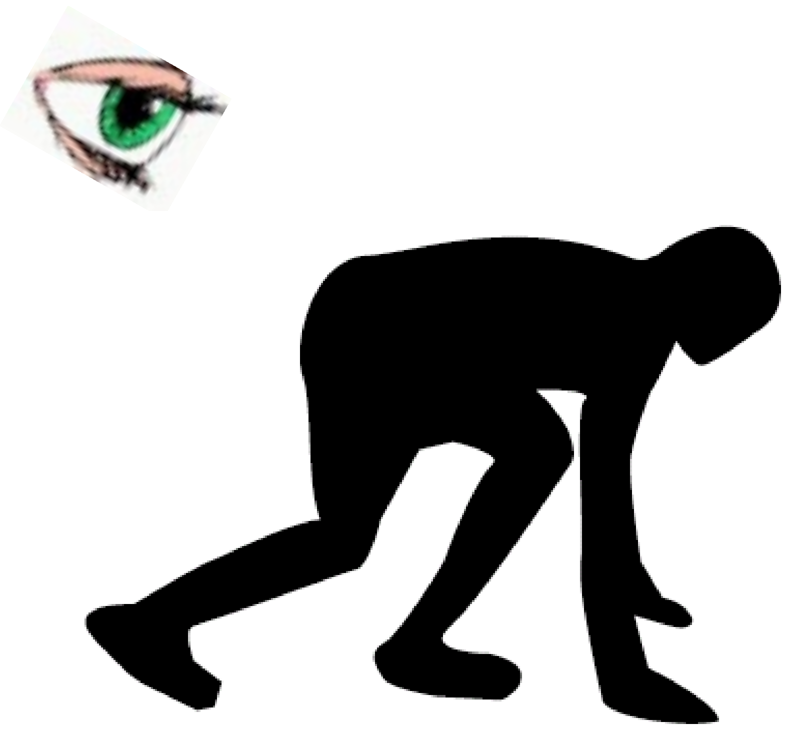


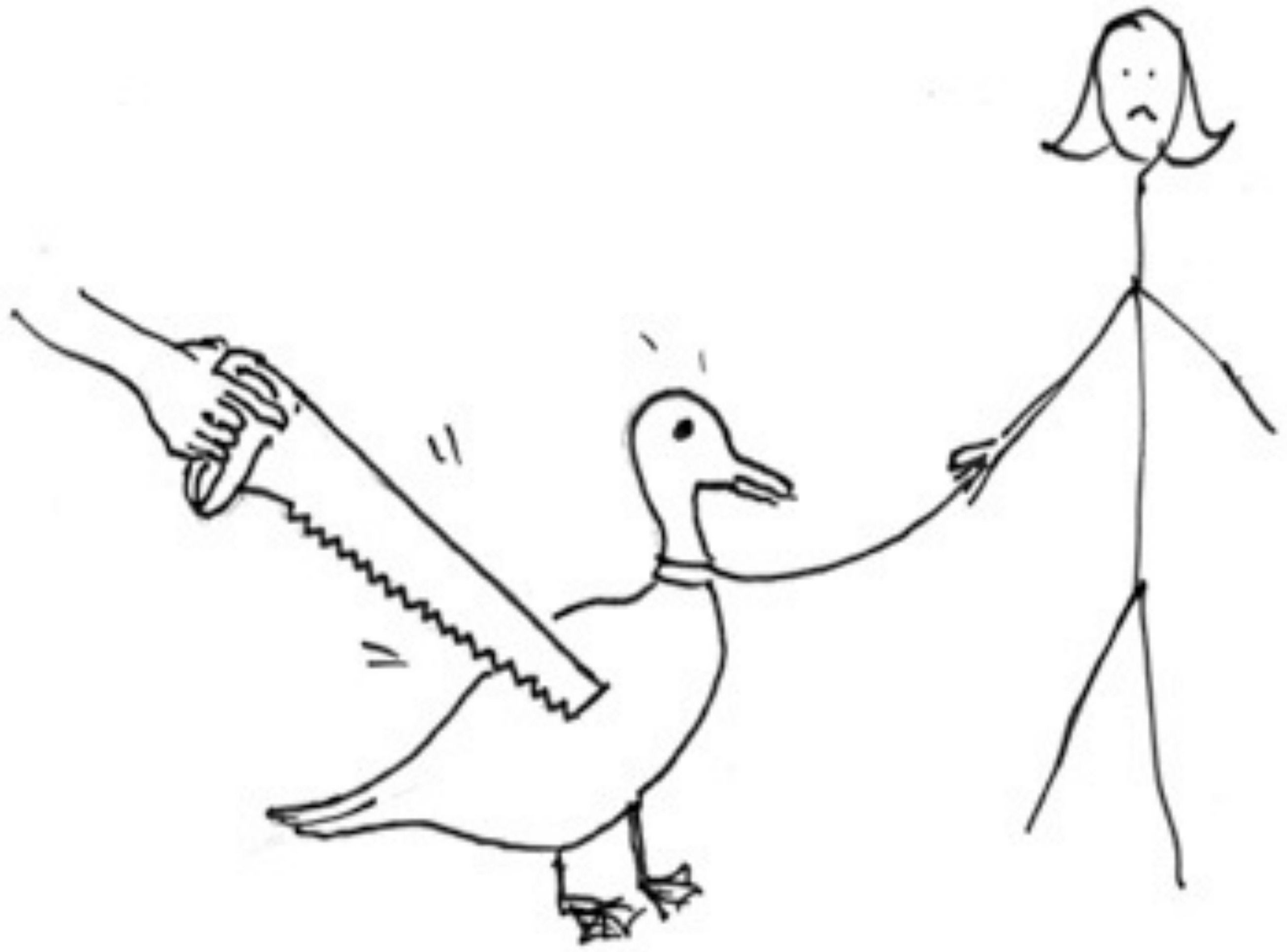
Aravind Joshi
(1929-2018)

I saw her duck

lexical ambiguity

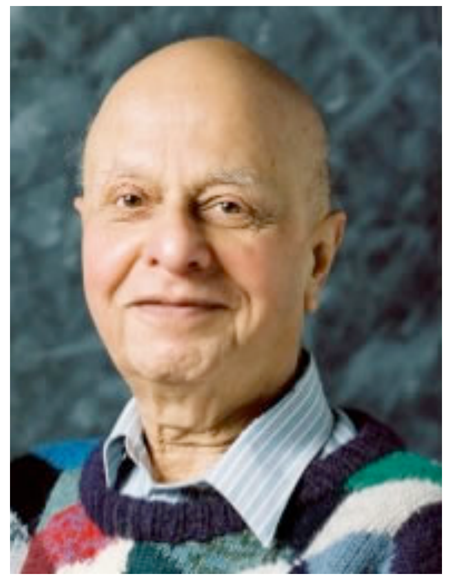






Language is Hard, Even for Humans

- *how many interpretations?*



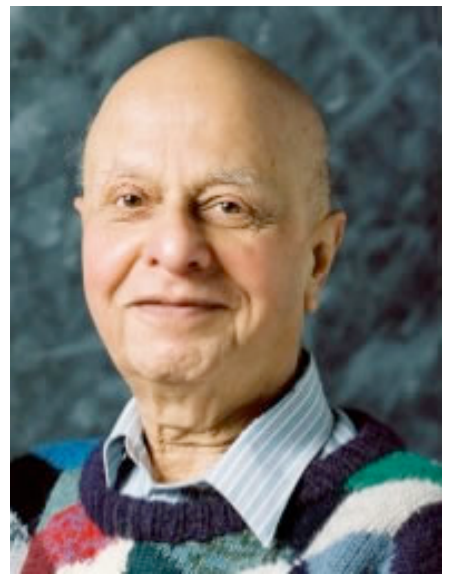
Aravind Joshi
(1929-2018)

I eat sushi with tuna

Language is Hard, Even for Humans

- *how many interpretations?*

I eat sushi with tuna



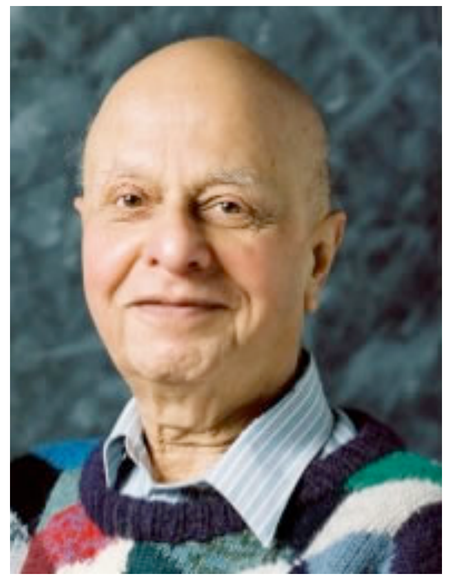
Aravind Joshi
(1929-2018)

Language is Hard, Even for Humans

- *how many interpretations?*

I eat sushi with tuna

structural ambiguity



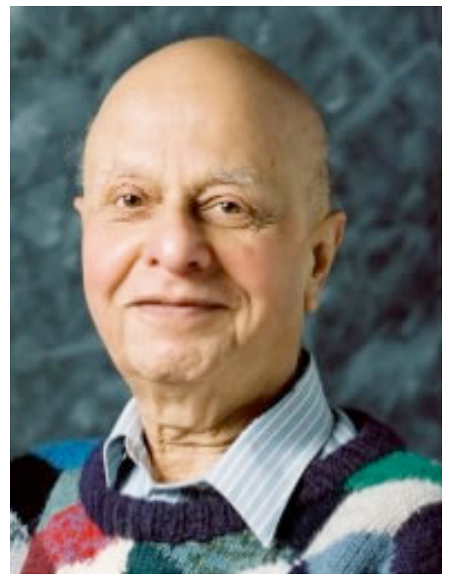
Aravind Joshi
(1929-2018)

Language is Hard, Even for Humans

- *how many interpretations?*

I eat sushi with tuna

structural ambiguity



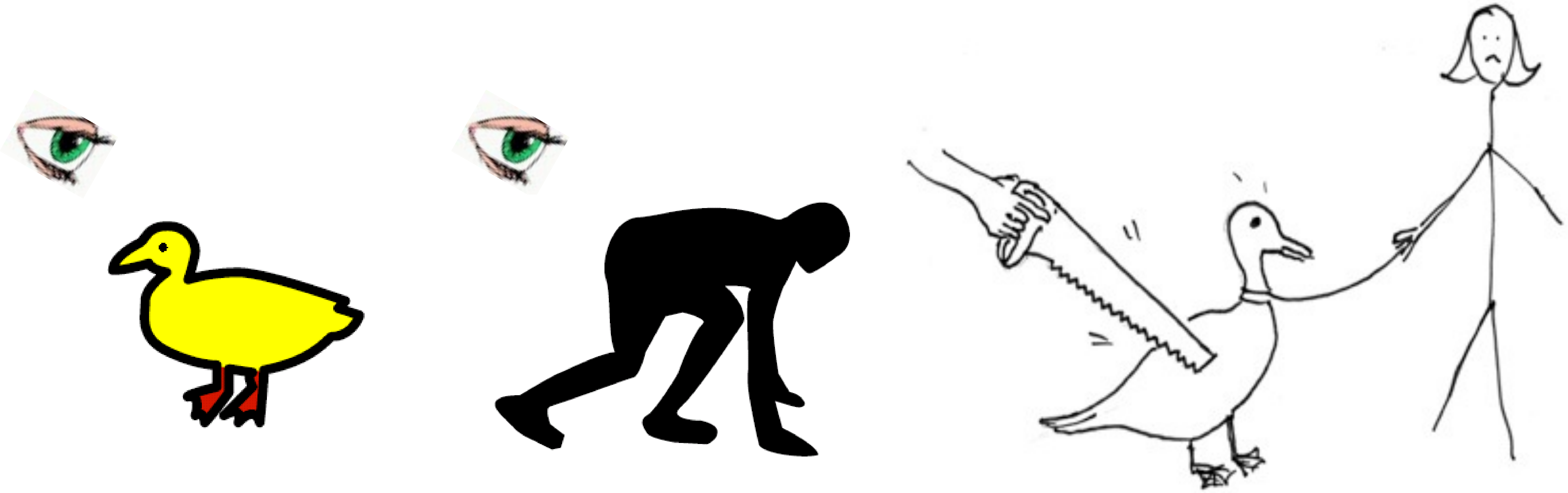
Aravind Joshi
(1929-2018)

Unexpected Structural Ambiguity

Language is Hard: Ambiguity Explosion

- *how many interpretations?*

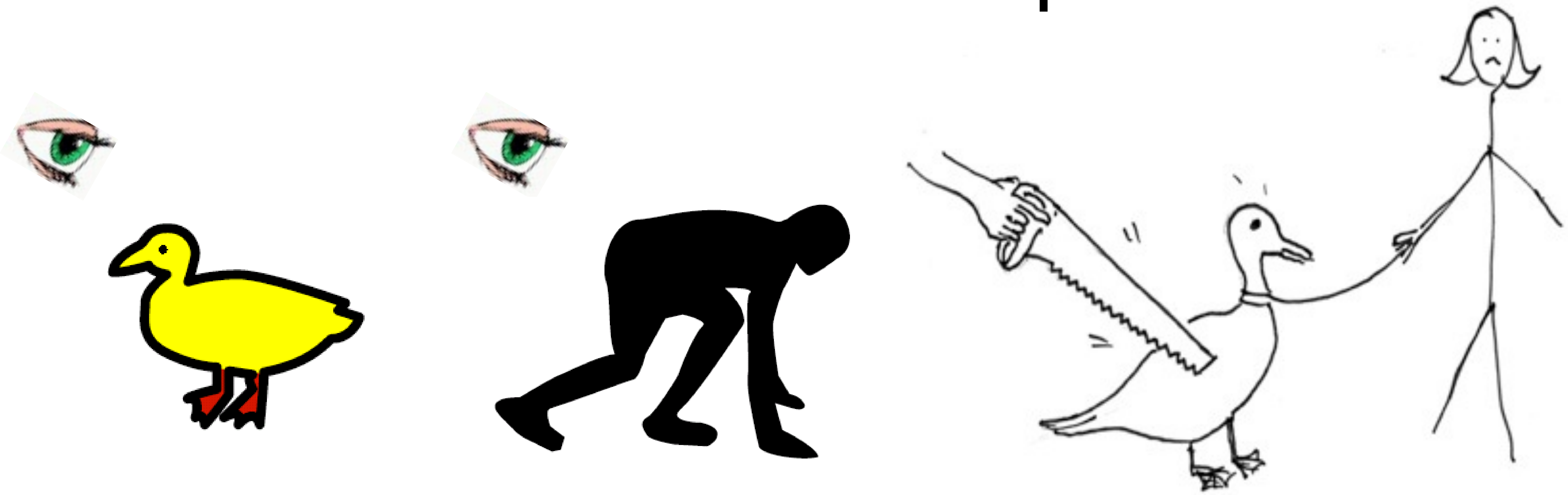
I saw her duck



Language is Hard: Ambiguity Explosion

- *how many interpretations?*

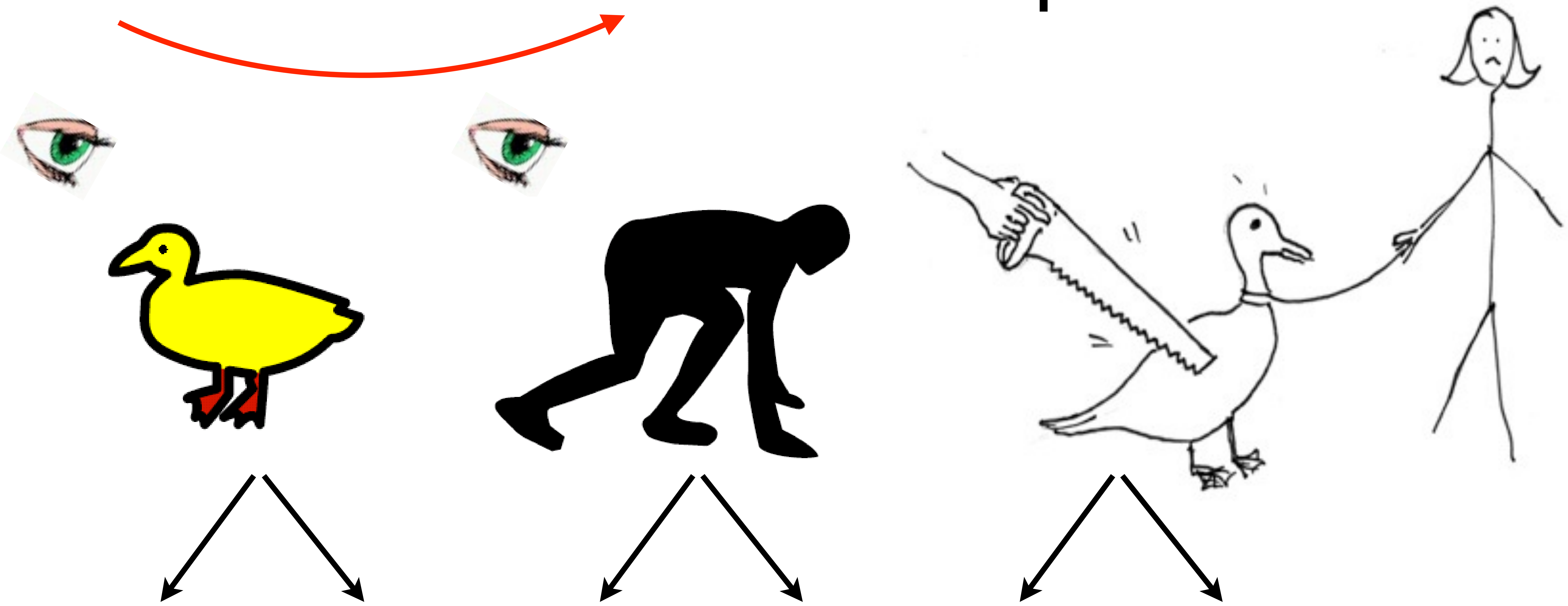
I saw her duck with a telescope



Language is Hard: Ambiguity Explosion

- *how many interpretations?*

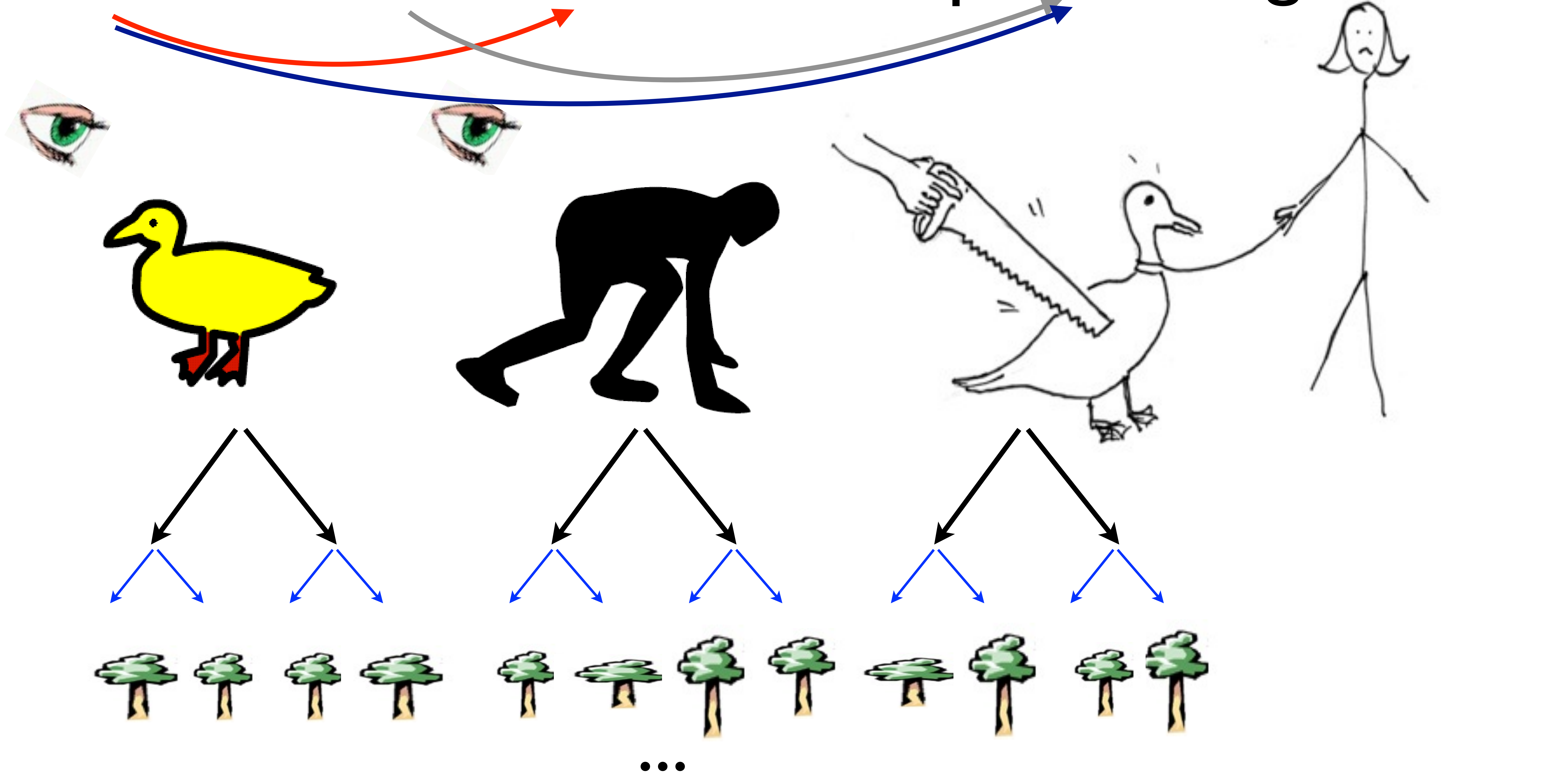
I saw her duck with a telescope



Language is Hard: Ambiguity Explosion

- *how many interpretations?*

I saw her duck with a telescope in the garden ...

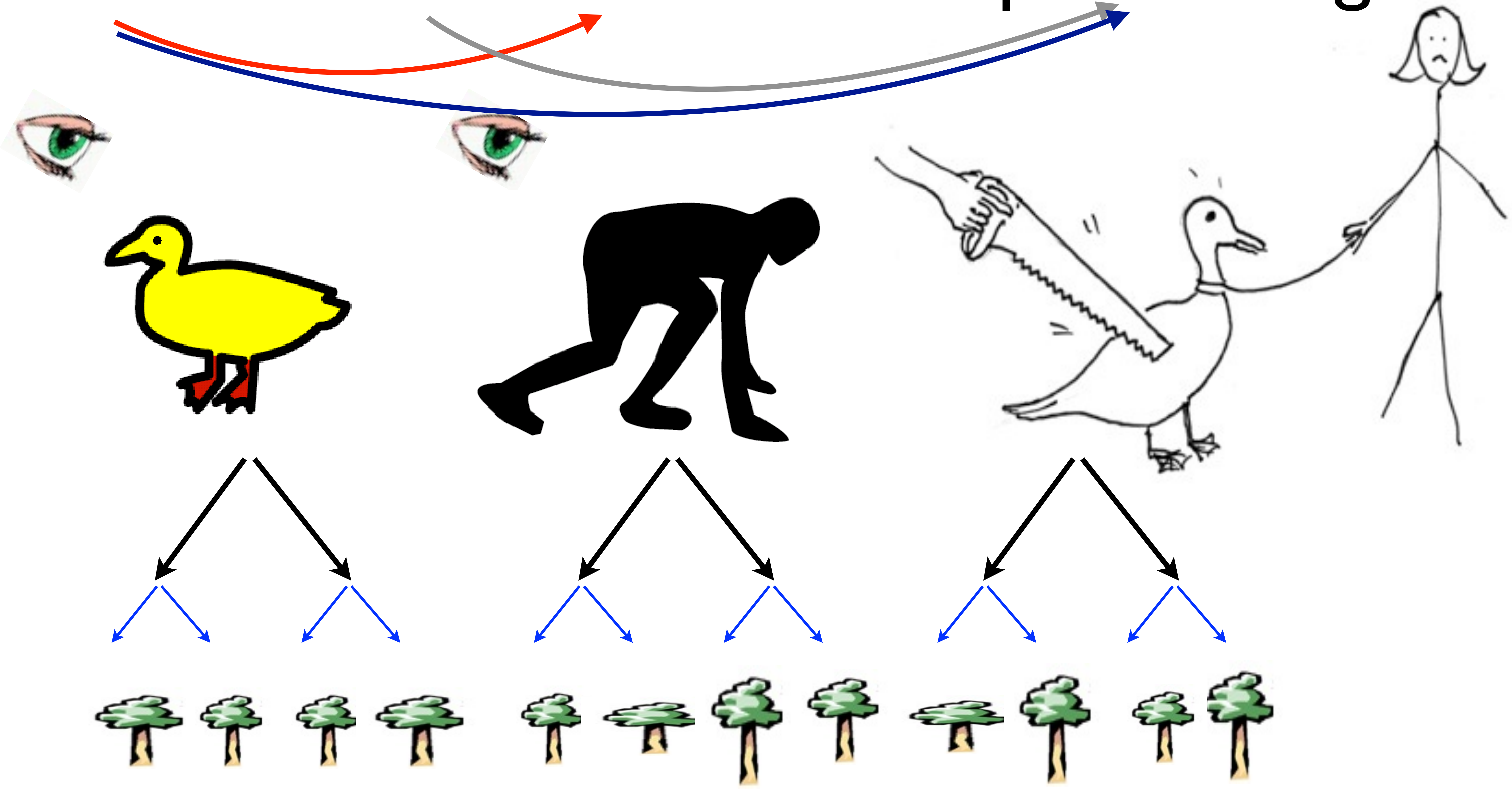


Language is Hard: Ambiguity Explosion

- *how many interpretations?*

But humans can resolve these ambiguities incremental in linear-time!

I saw her duck with a telescope in the garden ...



...

Human-Inspired NLP?

I eat sushi with tuna from Japan

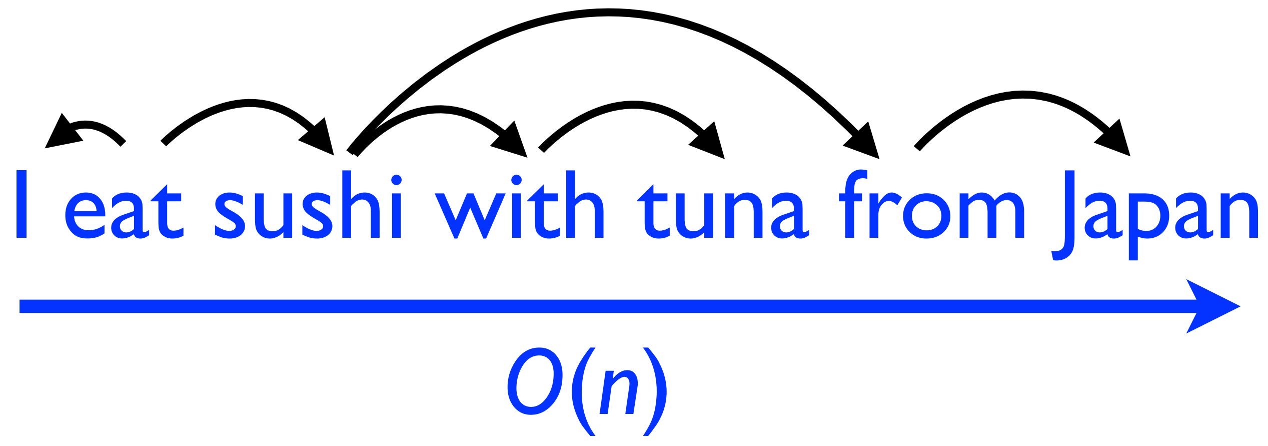
Human-Inspired NLP?

- human sentence processing is well-known to be incremental and linear-time

I eat sushi with tuna from Japan

Human-Inspired NLP?

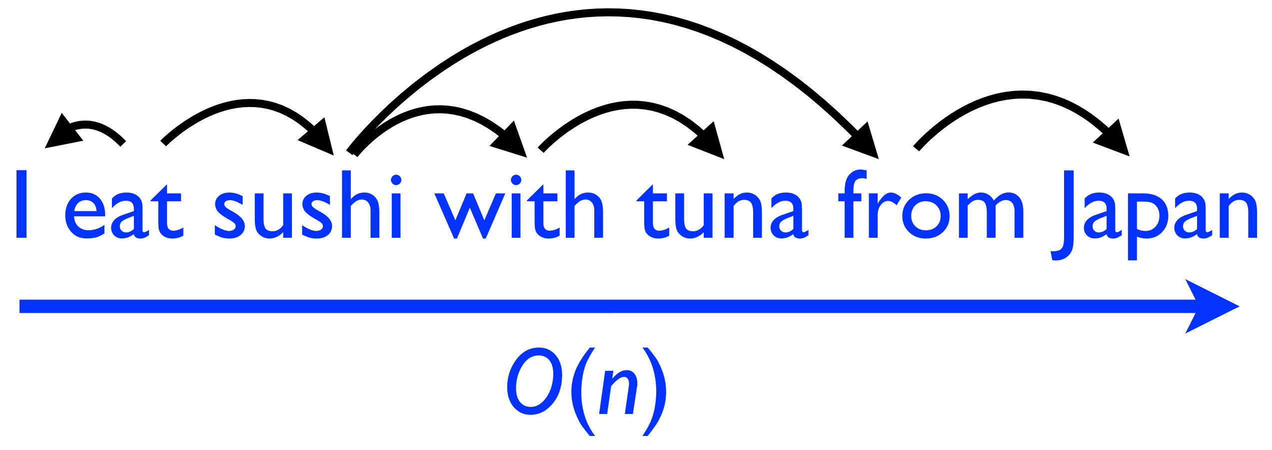
- human sentence processing is well-known to be incremental and linear-time



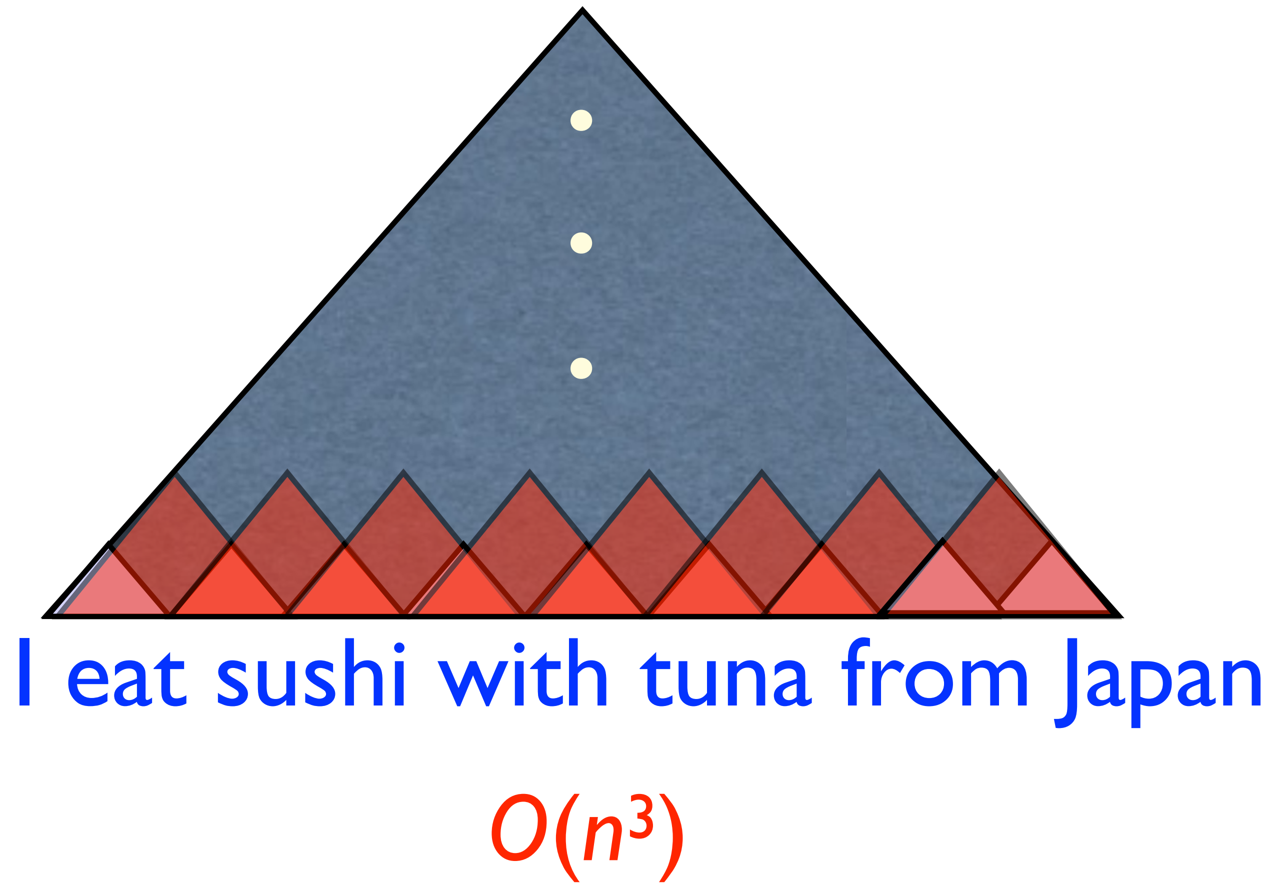
Human-Inspired NLP?

- human sentence processing is well-known to be incremental and linear-time



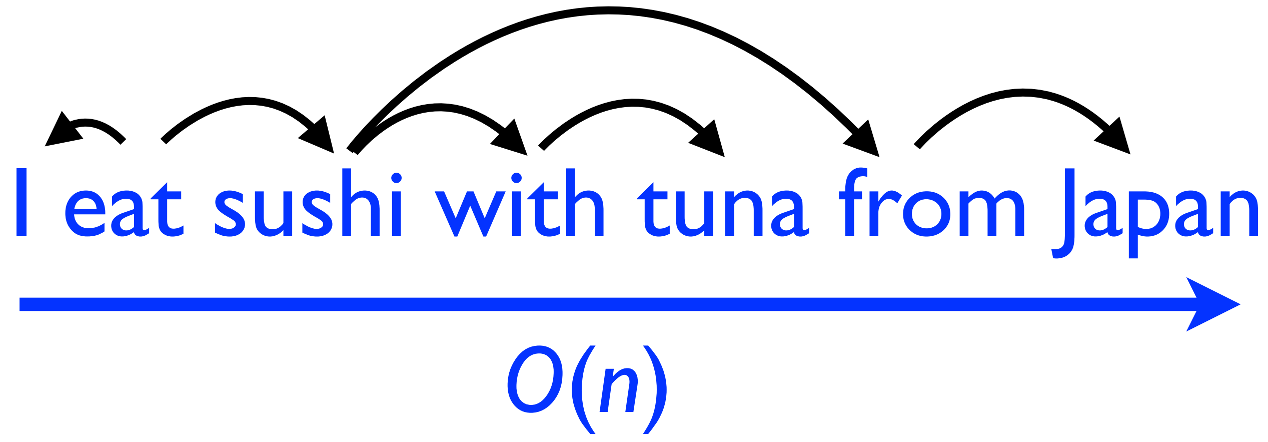


- natural language processing algorithms are often non-incremental and slow
  they often need full sentence as input and run in superlinear time

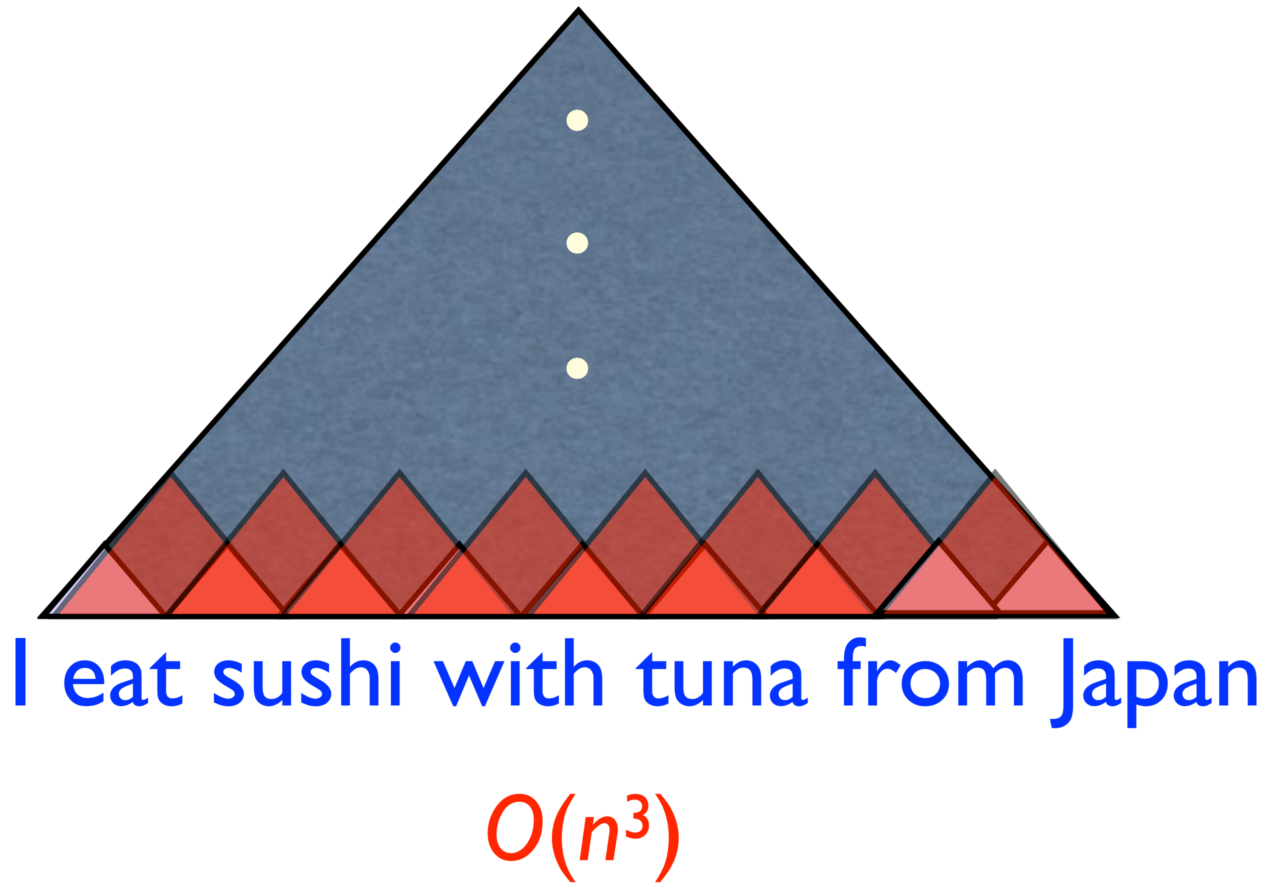


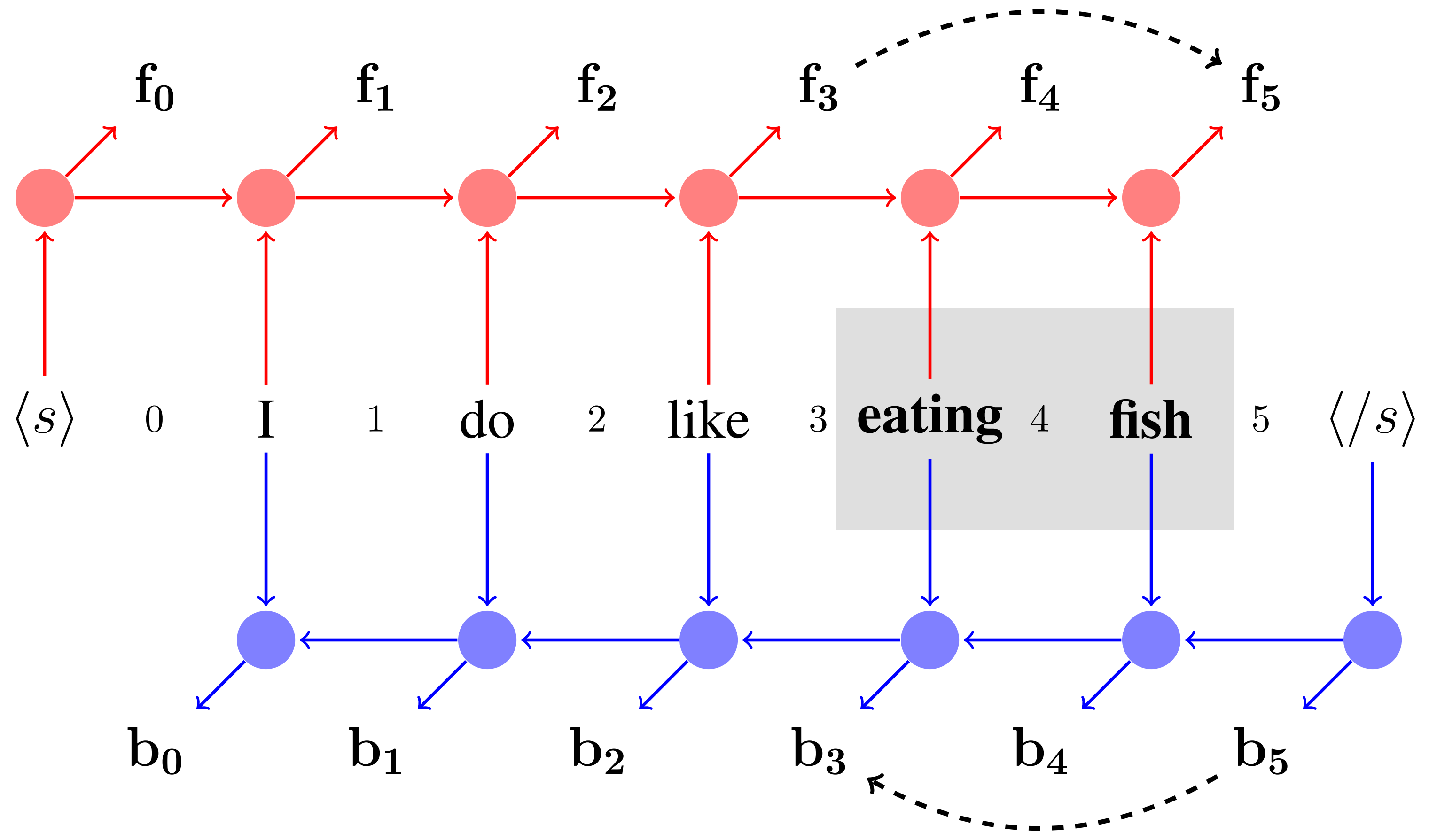
Human-Inspired NLP?

- human sentence processing is well-known to be incremental and linear-time



- natural language processing algorithms are often non-incremental and slow
  they often need full sentence as input and run in superlinear time

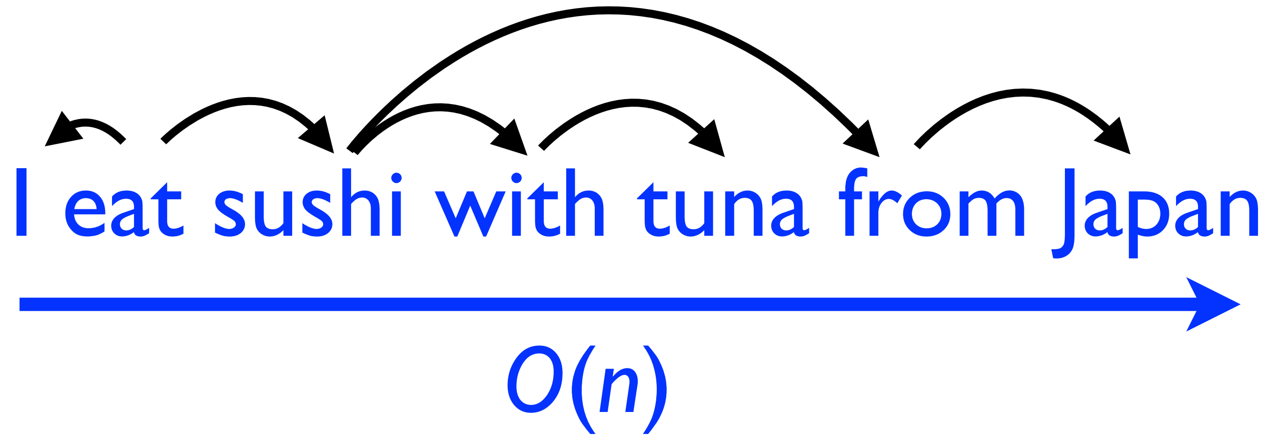




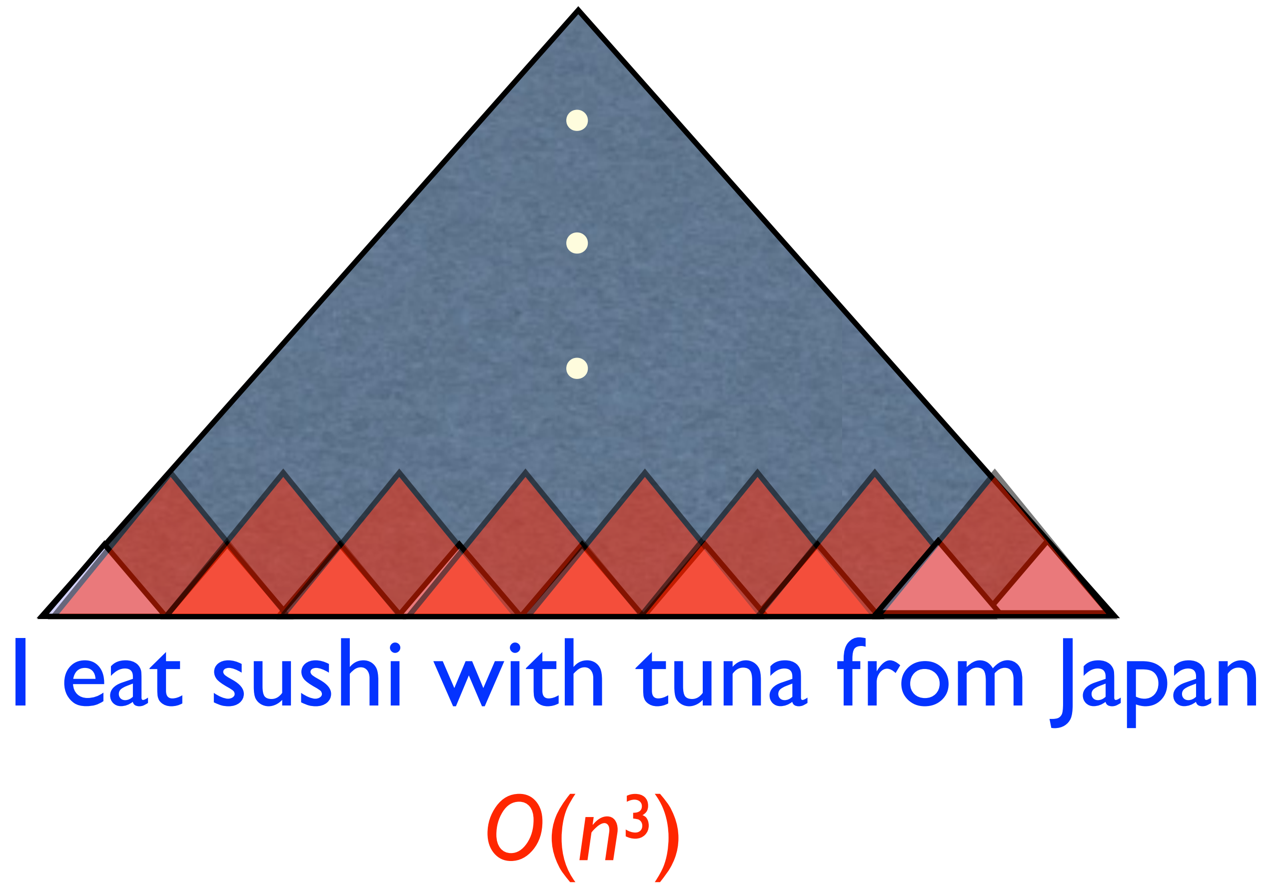
Human-Inspired NLP?

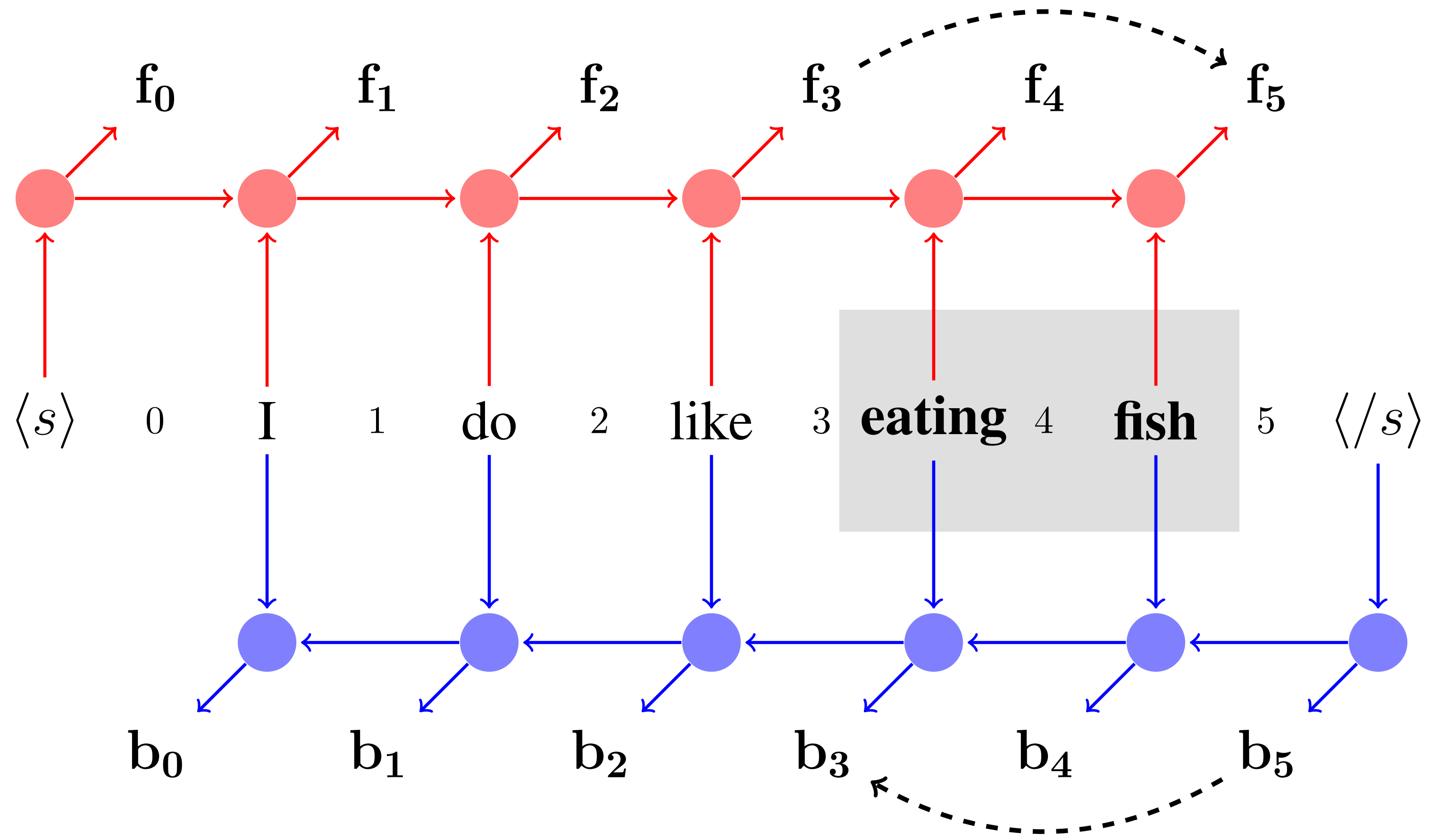
- human sentence processing is well-known to be incremental and linear-time

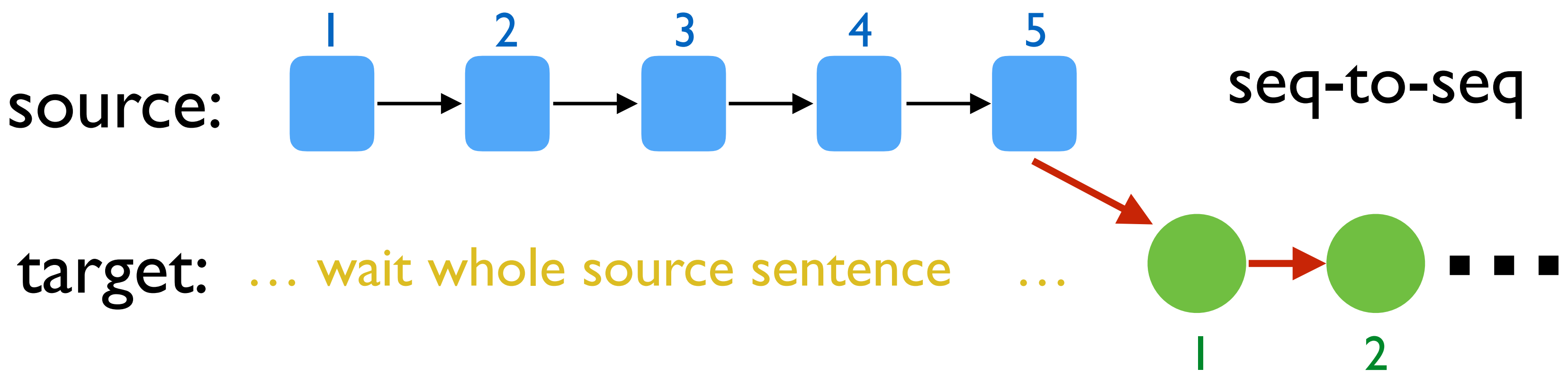




- natural language processing algorithms are often non-incremental and slow
  they often need full sentence as input and run in superlinear time







Three Stories on Incrementality and Instantaneity

simultaneous translation **incremental parsing** **linear-time RNA structure prediction**

Three Stories on Incrementality and Instantaneity

simultaneous translation	incremental parsing	linear-time RNA structure prediction
Bush met Putin in Moscow	I eat sushi with tuna from Japan	GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA
source language sentence	natural language sentence	RNA sequence

Three Stories on Incrementality and Instantaneity

simultaneous translation

target-language sequence

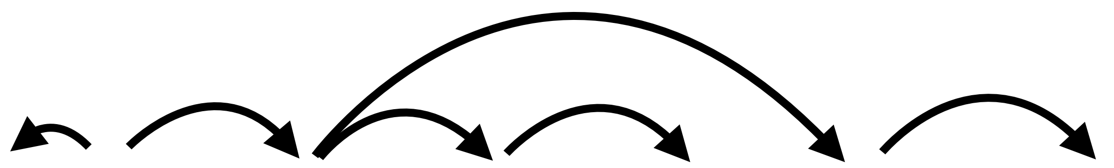
布什在莫斯科与普京会晤

Bush met Putin in Moscow

source language sentence

incremental parsing

syntactic structure

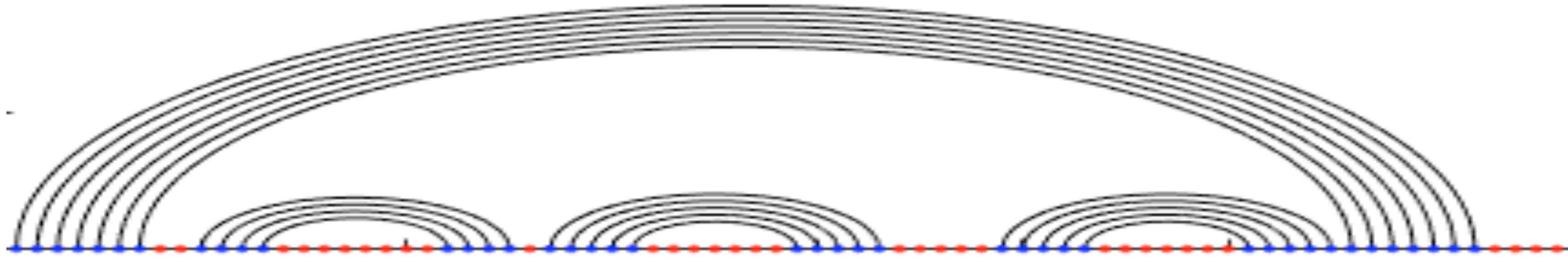


I eat sushi with tuna from Japan

natural language sentence

linear-time RNA structure prediction

secondary structure



GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

RNA sequence

Part 1: Simultaneous Translation

(Ma, Huang, et al, ArXiv 2018; under review)

Part 1: Simultaneous Translation

(Ma, Huang, et al, ArXiv 2018; under review)

Background: Consecutive vs. Simultaneous Interpretation

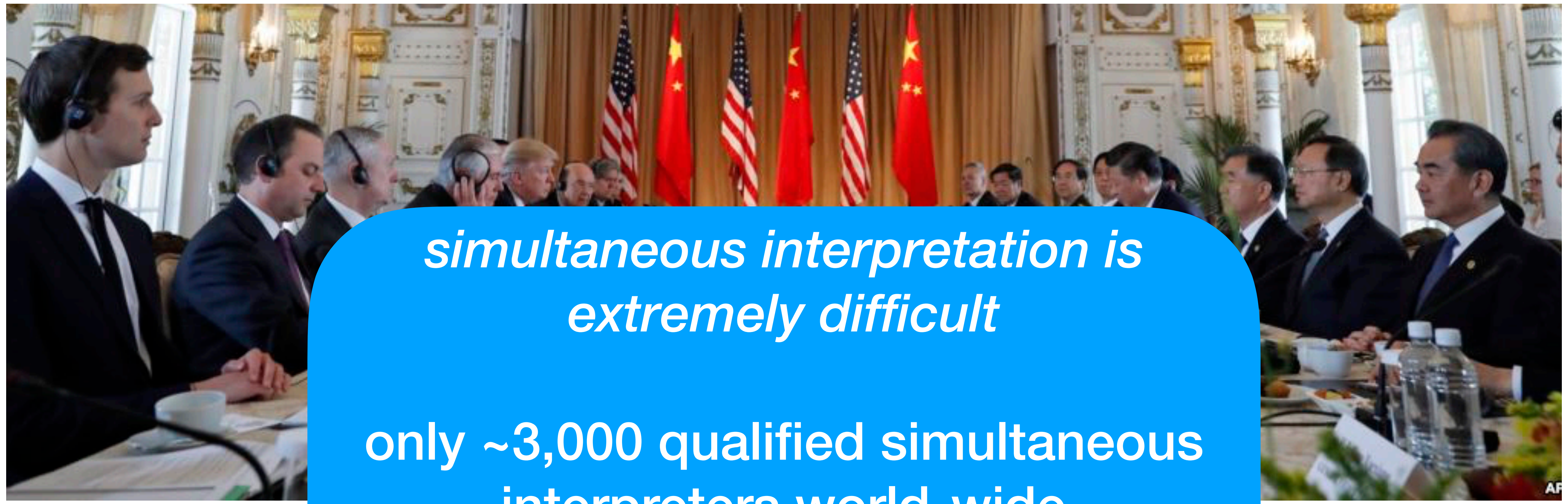
consecutive interpretation
multiplicative latency (x2)

simultaneous interpretation
additive latency (+3 secs)

Background: Consecutive vs. Simultaneous Interpretation

consecutive interpretation
multiplicative latency (x2)

simultaneous interpretation
additive latency (+3 secs)

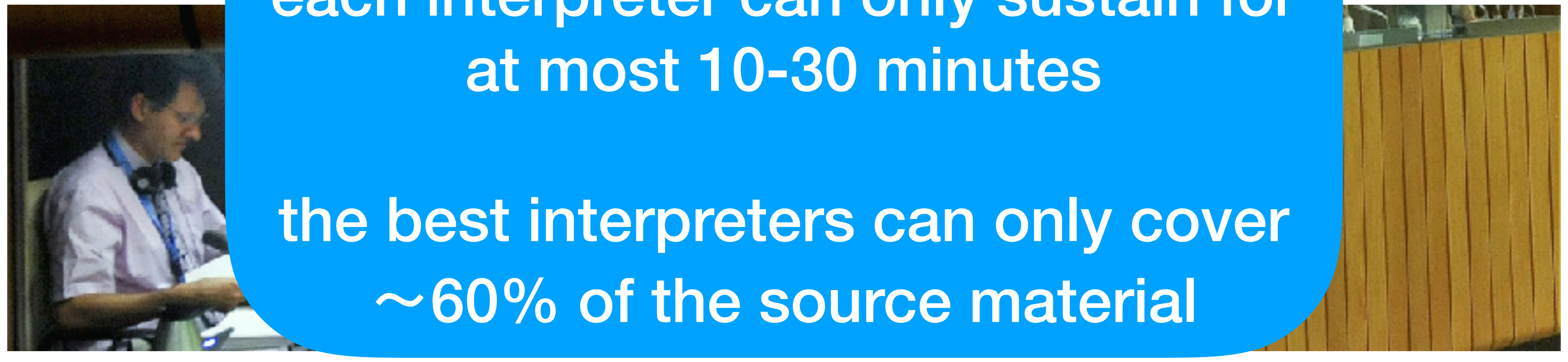


simultaneous interpretation is extremely difficult

only ~3,000 qualified simultaneous interpreters world-wide

each interpreter can only sustain for at most 10-30 minutes

the best interpreters can only cover ~60% of the source material



Background: Consecutive vs. Simultaneous Interpretation

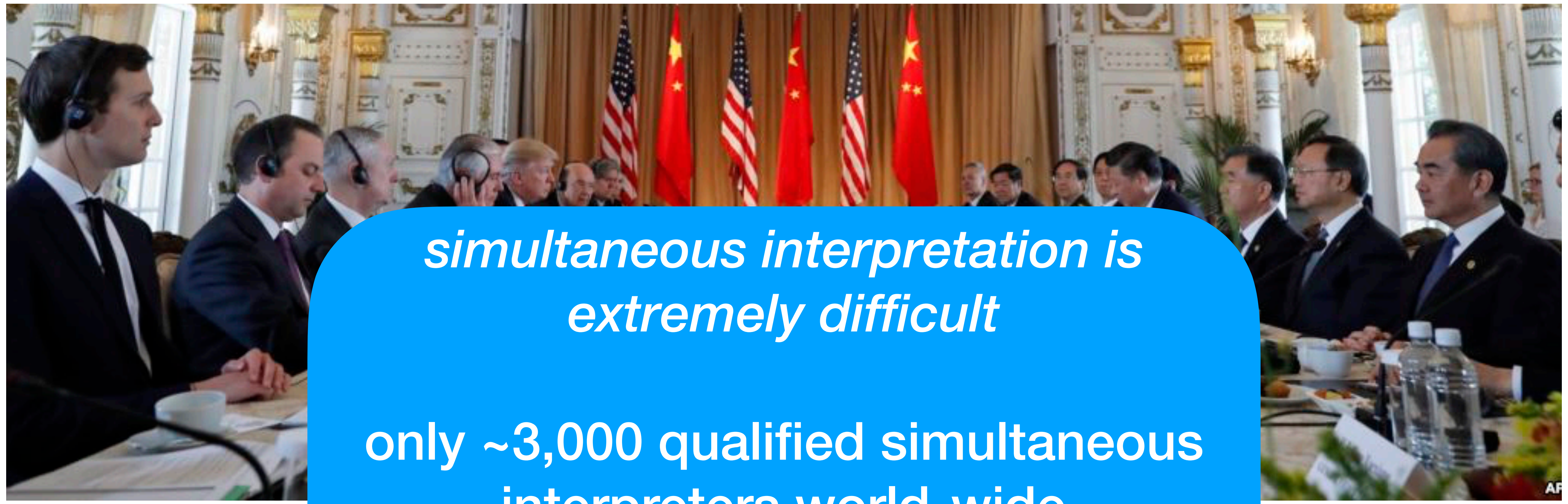
consecutive interpretation

multiplicative latency (x2)

just use standard
full-sentence translation
(e.g., seq-to-seq)

simultaneous interpretation

additive latency (+3 secs)

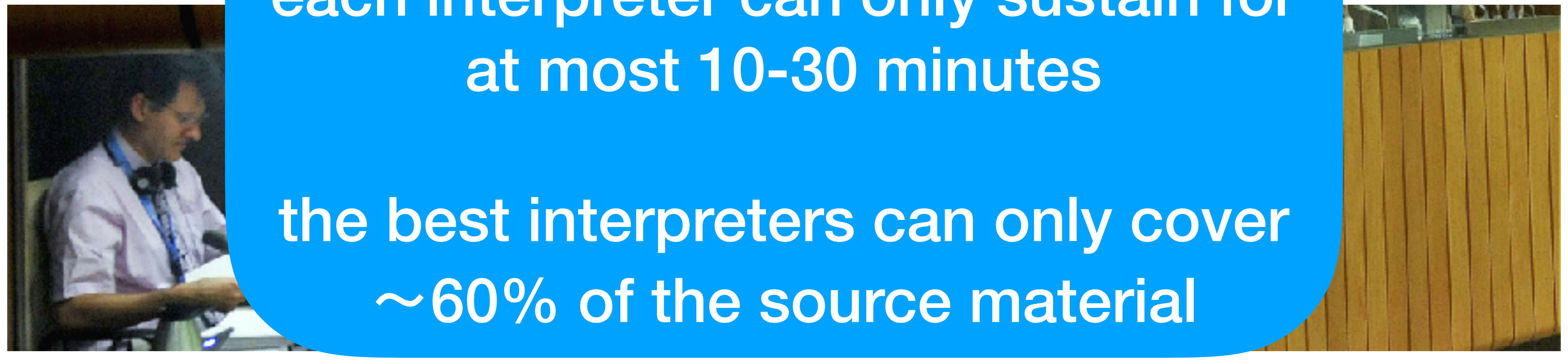


simultaneous interpretation is extremely difficult

only ~3,000 qualified simultaneous interpreters world-wide

each interpreter can only sustain for at most 10-30 minutes

the best interpreters can only cover ~60% of the source material

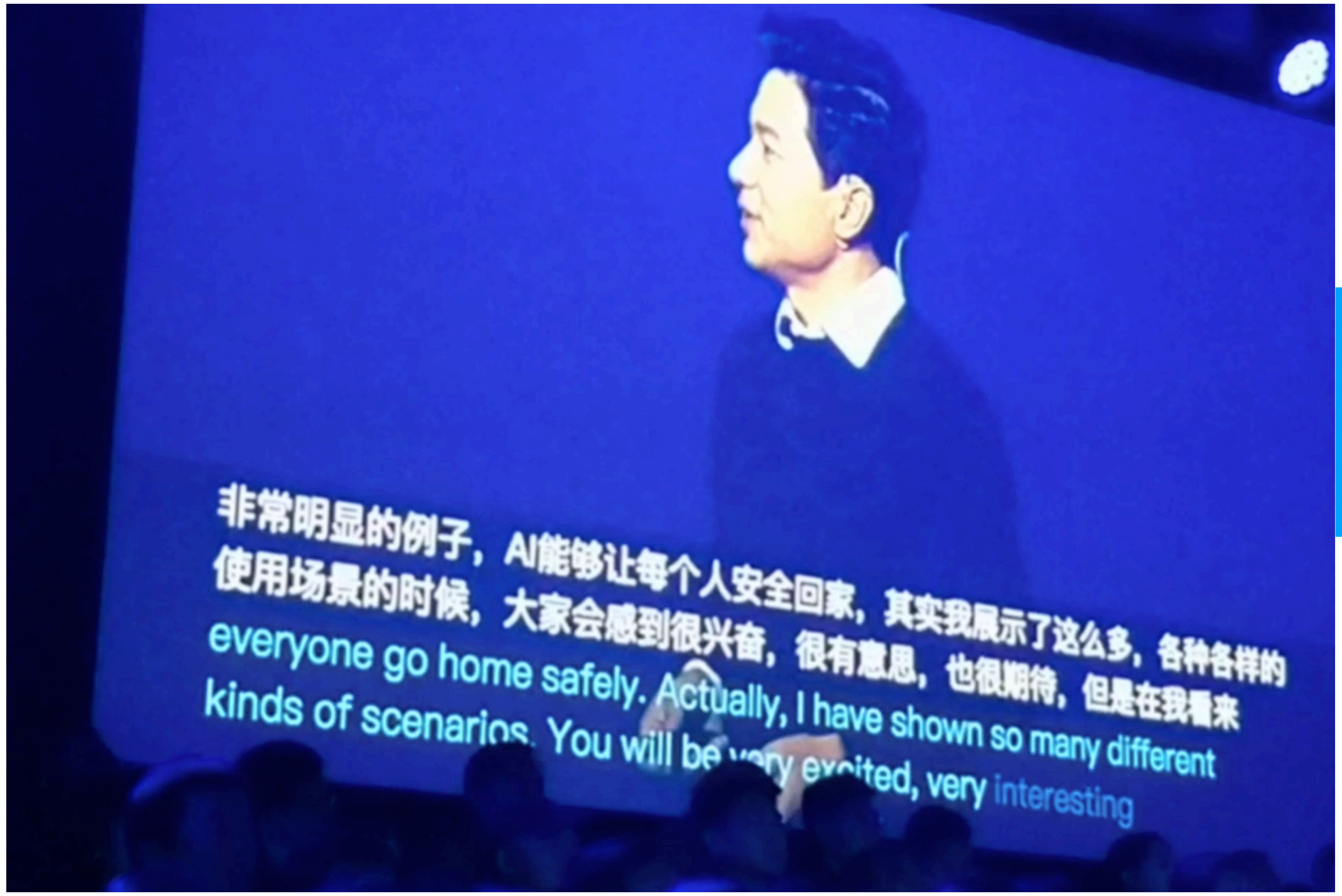


one of the holy grails of AI

need fundamentally different ideas!

Our Breakthrough

full-sentence (non-simultaneous) translation
latency: one sentence (10+ secs)



Baidu World Conference, November 2017

and many other companies

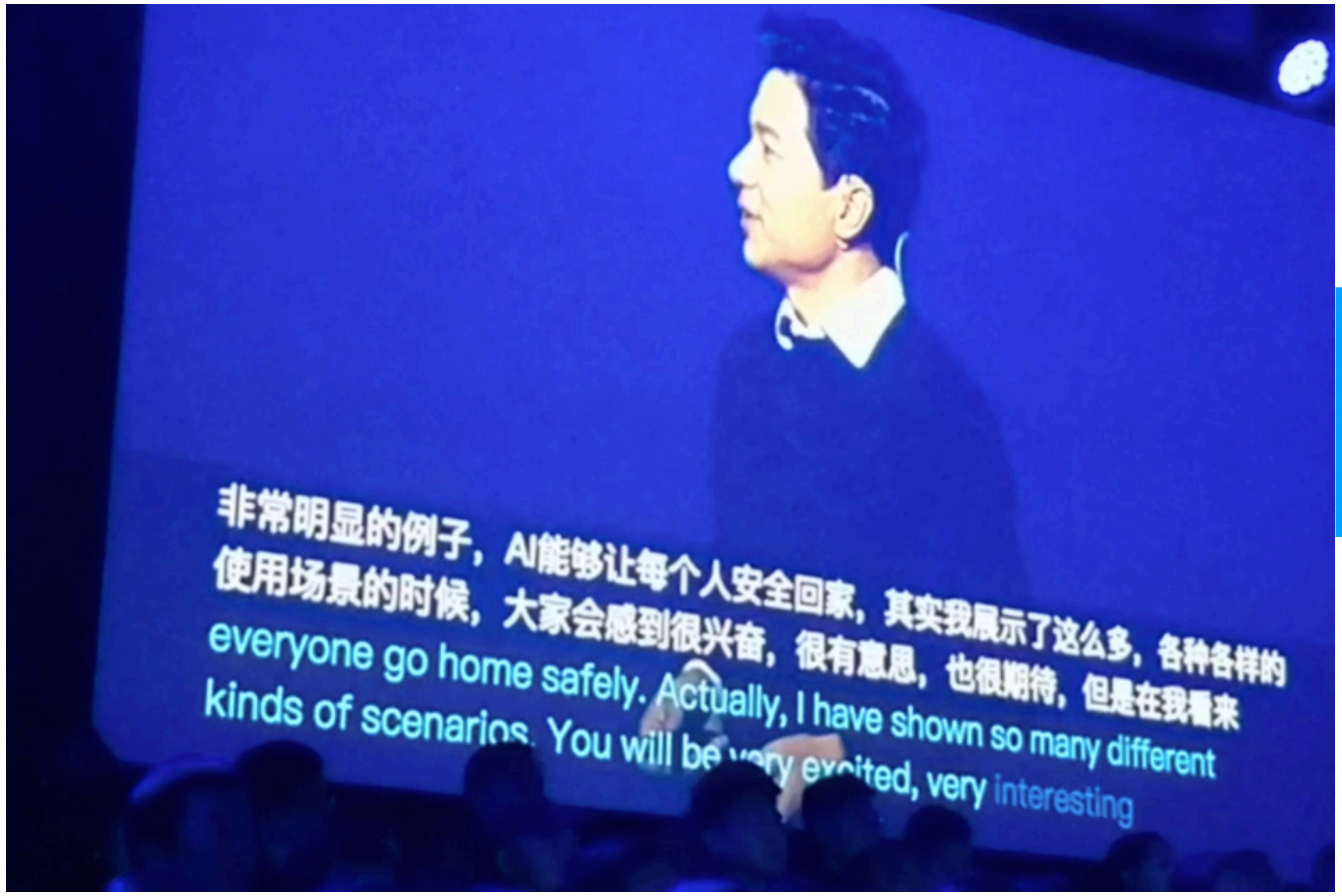
our
work

simultaneous translation achieved for the first time
latency ~3 secs

Baidu World Conference, November 2018

Our Breakthrough

full-sentence (non-simultaneous) translation
latency: one sentence (10+ secs)



Baidu World Conference, November 2017

and many other companies

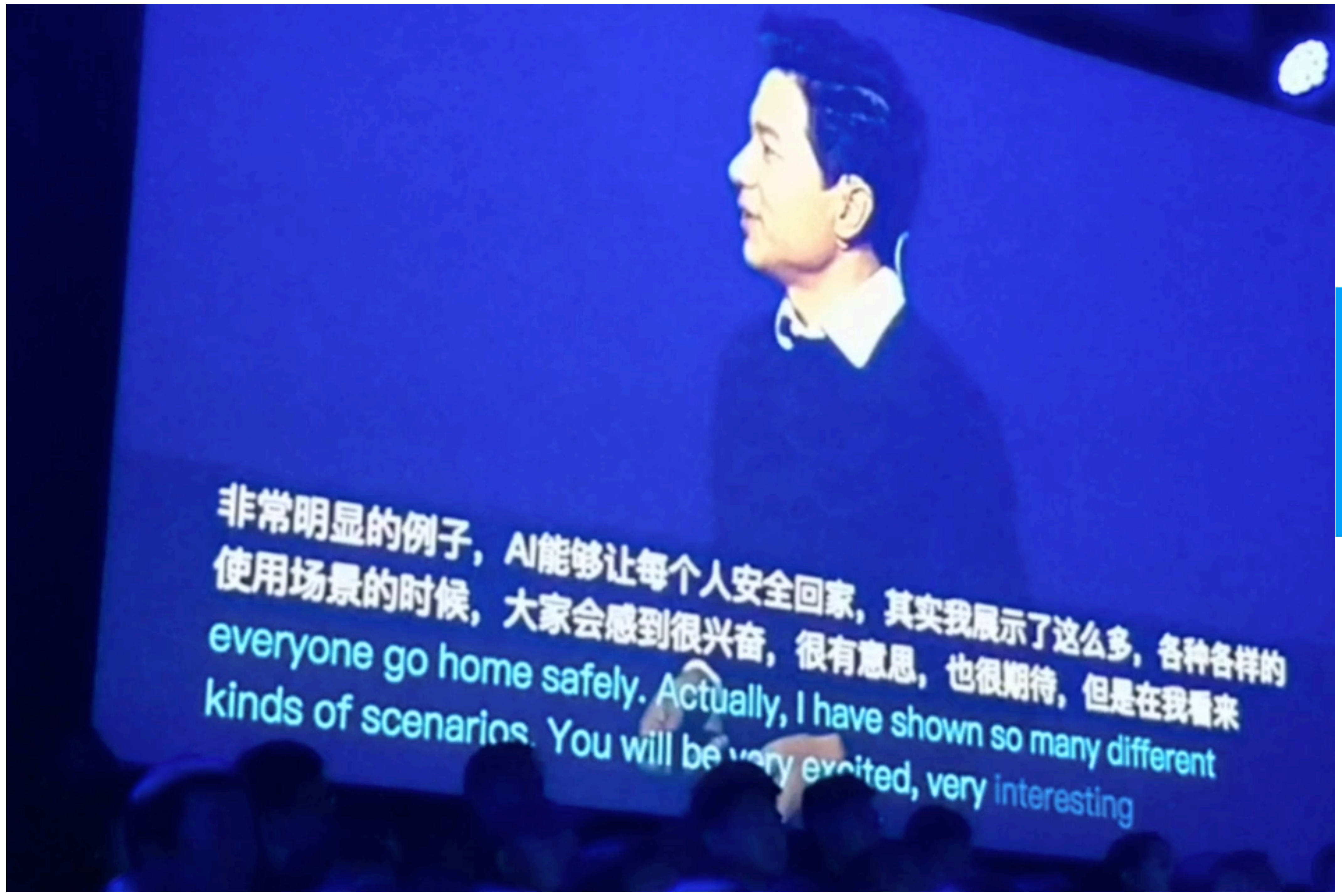
our work

simultaneous translation achieved for the first time
latency ~3 secs

Baidu World Conference, November 2018

Our Breakthrough

full-sentence (non-simultaneous) translation
latency: one sentence (10+ secs)



Baidu World Conference, November 2017

and many other companies

our work

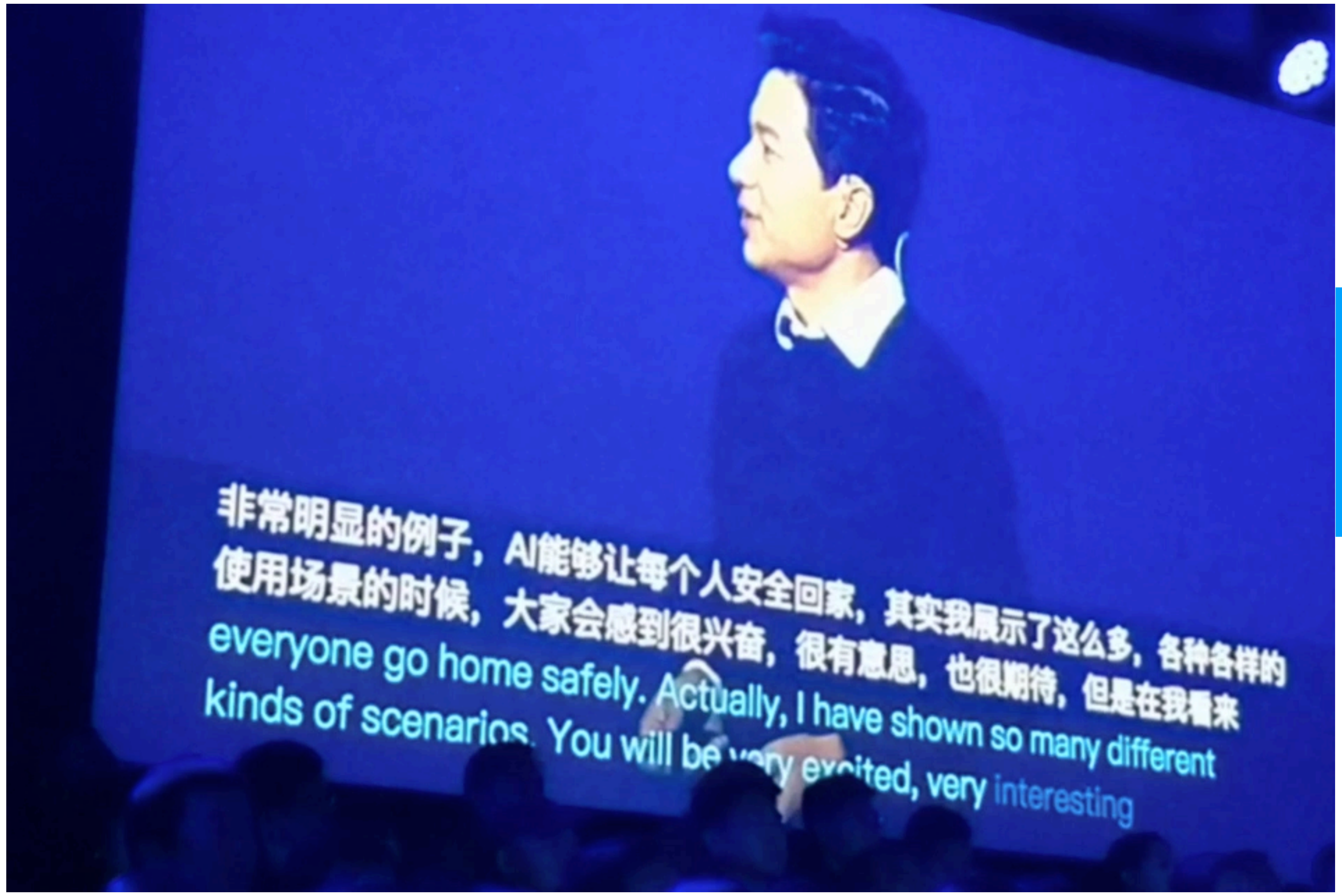
simultaneous translation achieved for the first time
latency ~3 secs

Baidu World Conference, November 2018

Our Breakthrough

full-sentence (non-simultaneous) translation
latency: one sentence (10+ secs)

simultaneous translation achieved for the first time
latency ~3 secs



our work

Baidu World Conference, November 2017

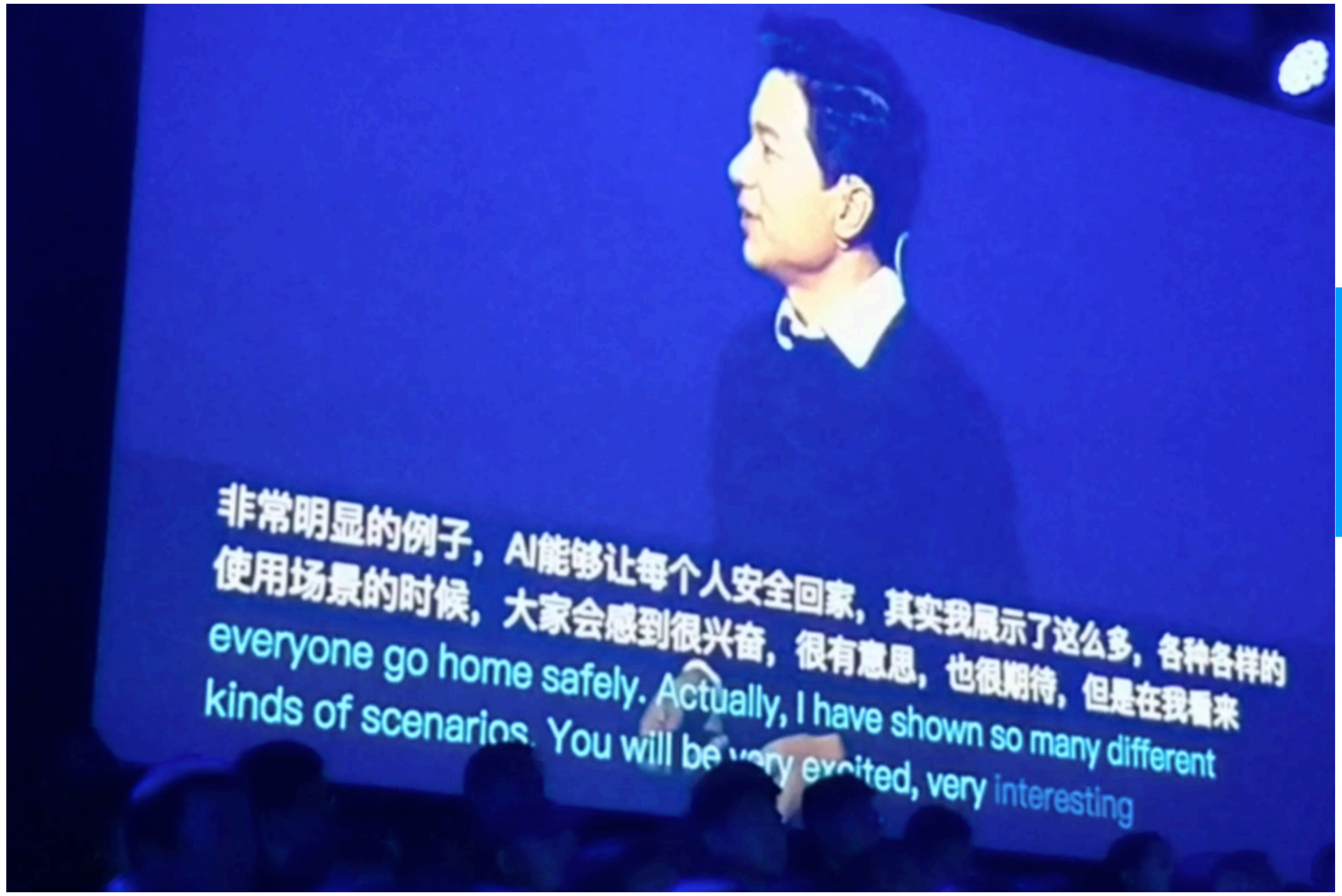
Baidu World Conference, November 2018

and many other companies

Our Breakthrough

full-sentence (non-simultaneous) translation
latency: one sentence (10+ secs)

simultaneous translation achieved for the first time
latency ~3 secs



our work

Baidu World Conference, November 2017

Baidu World Conference, November 2018

and many other companies

Challenge: Word Order Difference

- e.g. translate from Subj-Obj-Verb (Japanese, German) to Subj-Verb-Obj (English)
  - German is underlyingly SOV, and Chinese is a mix of SVO and SOV
  - human simultaneous interpreters routinely "anticipate" (e.g., predicting German verb)

ich	bin	mit	dem	Zug	nach	Ulm	**gefahren**
I	am	with	the	train	to	Ulm	**traveled**
I		(. . . . . . *waiting*	. . . . . .)				**traveled** by train to Ulm

Grissom et al, 2014

Challenge: Word Order Difference

- e.g. translate from Subj-Obj-Verb (Japanese, German) to Subj-Verb-Obj (English)
  - German is underlyingly SOV, and Chinese is a mix of SVO and SOV
  - human simultaneous interpreters routinely “anticipate” (e.g., predicting German verb)

ich	bin	mit	dem	Zug	nach	Ulm	**gefahren**
I	am	with	the	train	to	Ulm	**traveled**
I	*(. . . . . . waiting. . . . . .)*						**traveled** by train to Ulm

Grissom et al, 2014

Bùshí	*zǒngtǒng*	*zài*	*Mòsīkē*	*yǔ*	*Éluósī*	*zǒngtǒng*	*Pǔjīng*	*huìwù*
布什	总统	在	莫斯科	与	俄罗斯	总统	普京	会晤
Bush	*President*	*in*	*Moscow*	*with*	*Russian*	*President*	*Putin*	*meet*

President Bush meets with Russian President Putin in Moscow

Challenge: Word Order Difference

- e.g. translate from Subj-Obj-Verb (Japanese, German) to Subj-Verb-Obj (English)
  - German is underlyingly SOV, and Chinese is a mix of SVO and SOV
  - human simultaneous interpreters routinely "anticipate" (e.g., predicting German verb)

ich	bin	mit	dem	Zug	nach	Ulm	**gefahren**
I	am	with	the	train	to	Ulm	**traveled**

I (. . . . . . *waiting*. . . . . .) **traveled** by train to Ulm

Grissom et al, 2014

Bùshí	zǒngtǒng	zài	Mòsīkē	yǔ	Éluósī	zǒngtǒng	Pǔjīng	huìwù
布什	总统	在	莫斯科	与	俄罗斯	总统	普京	会晤
Bush	*President*	*in*	*Moscow*	*with*	*Russian*	*President*	*Putin*	*meet*

President Bush meets with Russian President Putin in Moscow

non-anticipative: President Bush (…… *waiting* ……) meets with Russian …

Challenge: Word Order Difference

- e.g. translate from Subj-Obj-Verb (Japanese, German) to Subj-Verb-Obj (English)
  - German is underlyingly SOV, and Chinese is a mix of SVO and SOV
  - human simultaneous interpreters routinely "anticipate" (e.g., predicting German verb)

ich	bin	mit	dem	Zug	nach	Ulm	**gefahren**
I	am	with	the	train	to	Ulm	**traveled**

I (. . . . . . *waiting*. . . . . .) **traveled** by train to Ulm

Grissom et al, 2014

Bùshí	zǒngtǒng	zài	Mòsīkē	yǔ	Éluósī	zǒngtǒng	Pǔjīng	huìwù
布什	总统	在	莫斯科	与	俄罗斯	总统	普京	会晤
Bush	*President*	*in*	*Moscow*	*with*	*Russian*	*President*	*Putin*	*meet*

President Bush meets with Russian President Putin in Moscow

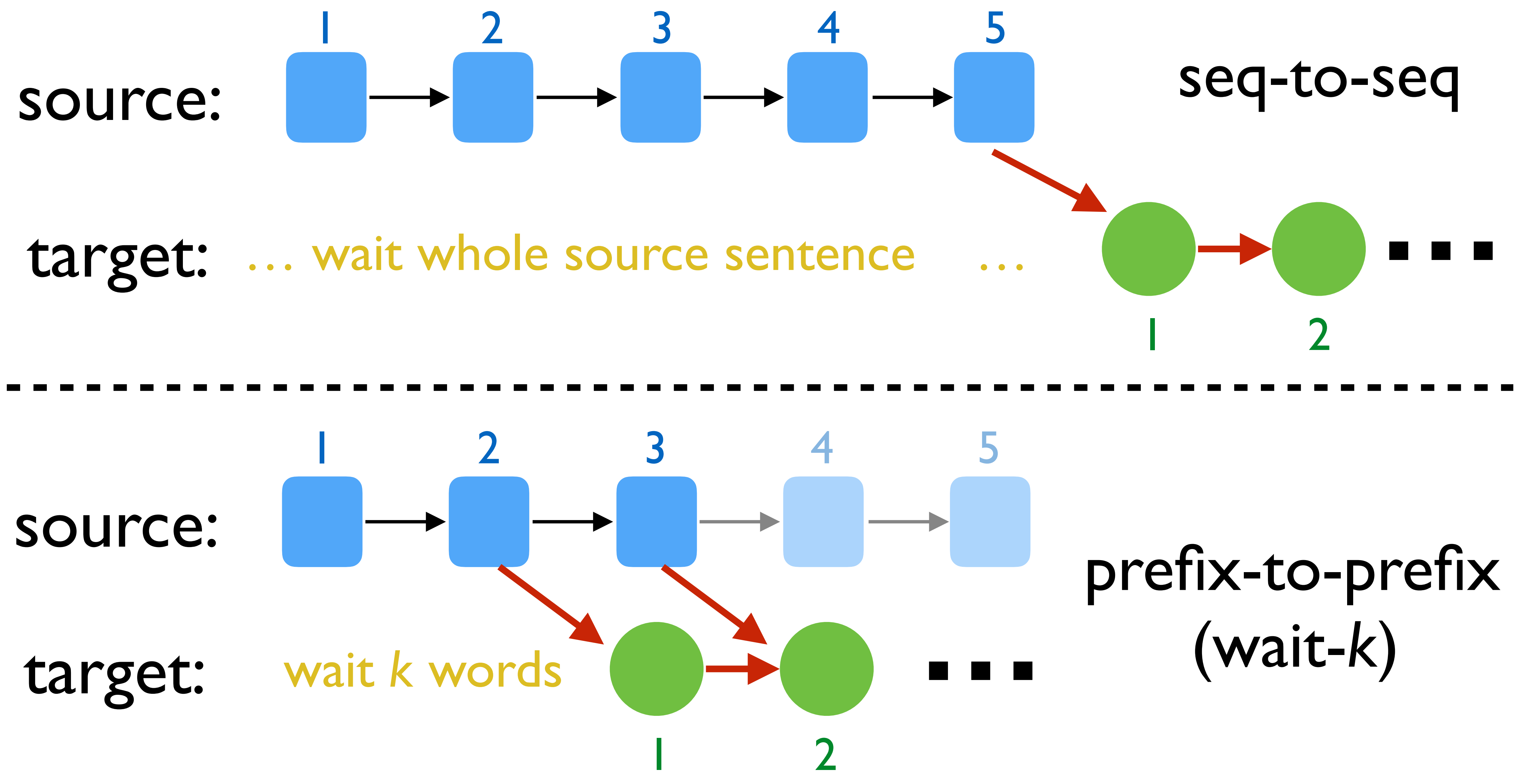
non-anticipative: President Bush (…… *waiting* ……) meets with Russian …

anticipative: President Bush meets with Russian President Putin in Moscow

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-k policy: translation is always k words behind source sentence
  - training in this way enables anticipation

$p(y_i \mid x_1 \ldots x_n, y_1 \ldots y_{i-1})$

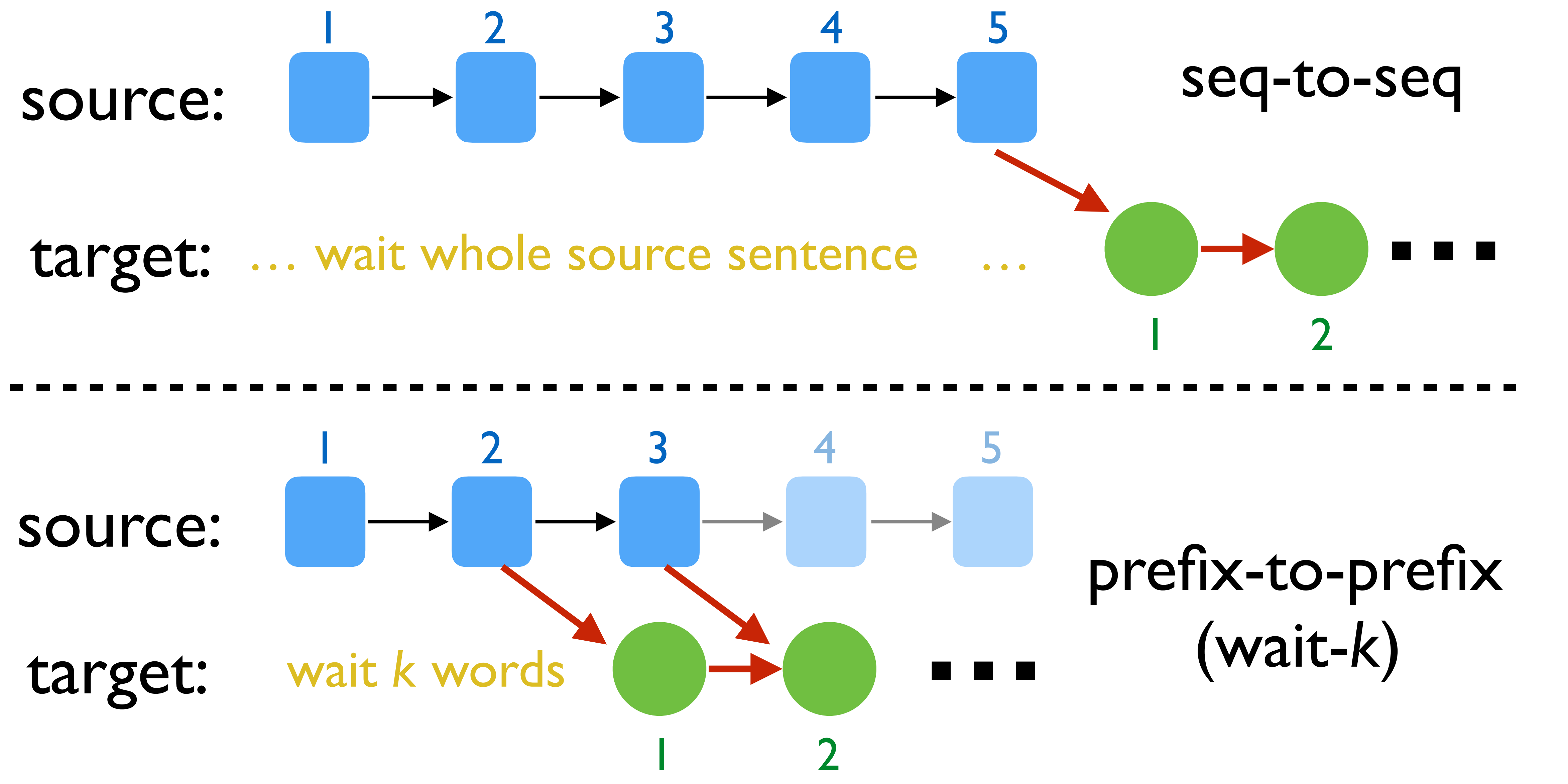


$p(y_i \mid x_1 \ldots x_{i+k-1}, y_1 \ldots y_{i-1})$

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

$$p(y_i \mid x_1 \ldots x_n, y_1 \ldots y_{i-1})$$



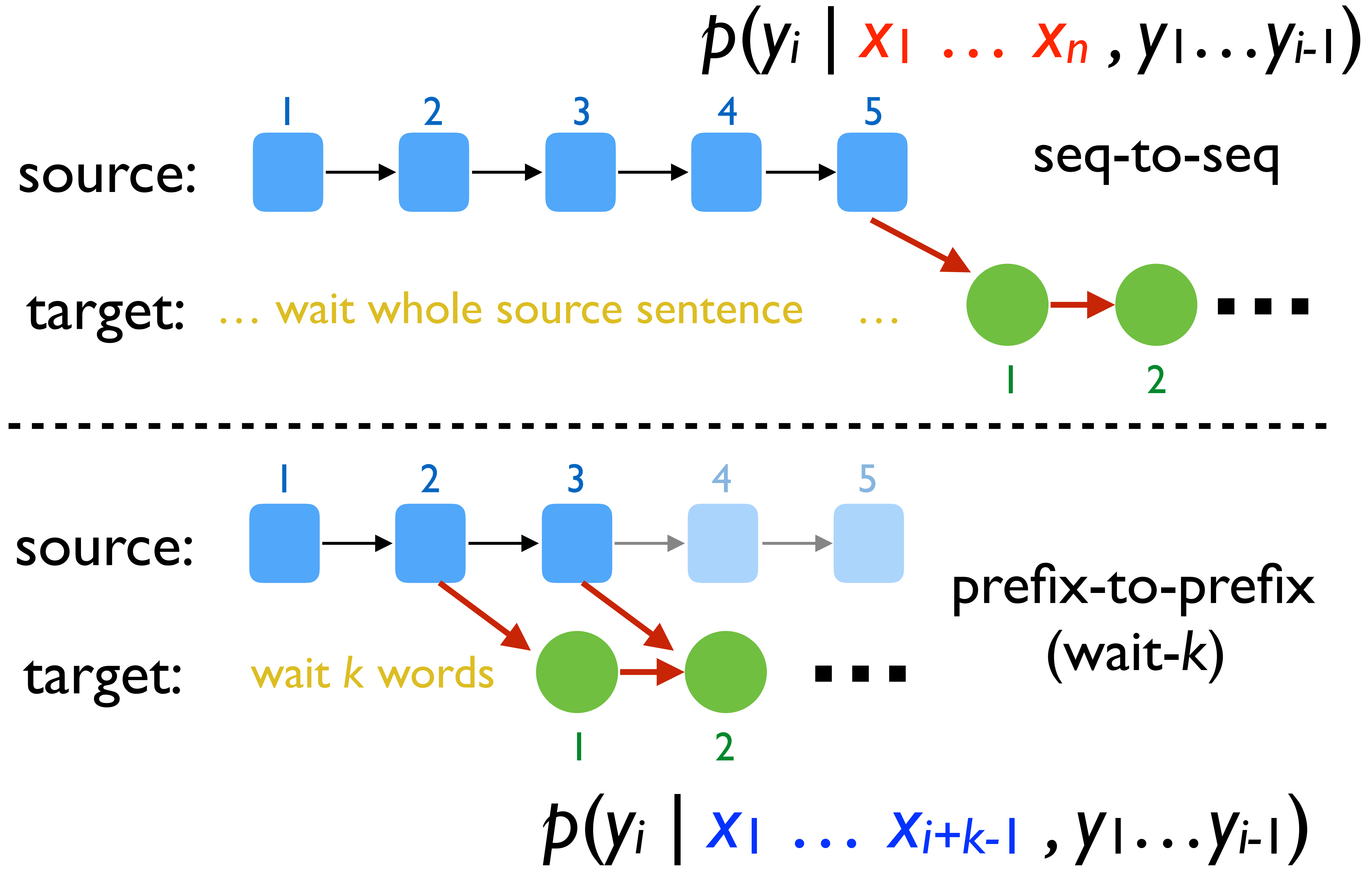
$$p(y_i \mid x_1 \ldots x_{i+k-1}, y_1 \ldots y_{i-1})$$

Bùshí	*zǒngtǒng*
布什	总统
Bush	*President*

wait 2 President

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-k policy: translation is always k words behind source sentence
  - training in this way enables anticipation

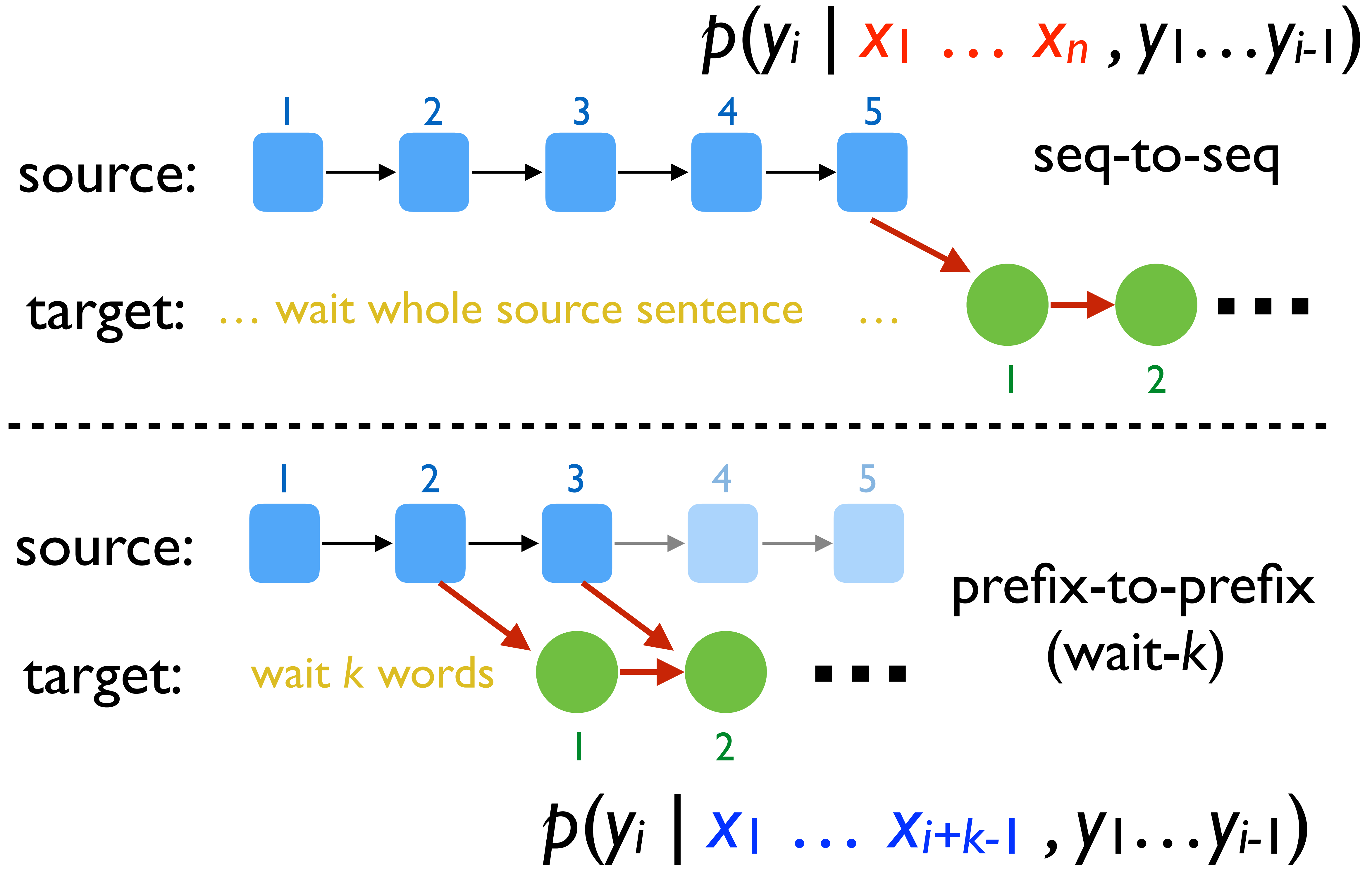


Bùshí	zǒngtǒng	zài
布什	总统	在
Bush	President	in

wait 2 President Bush

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

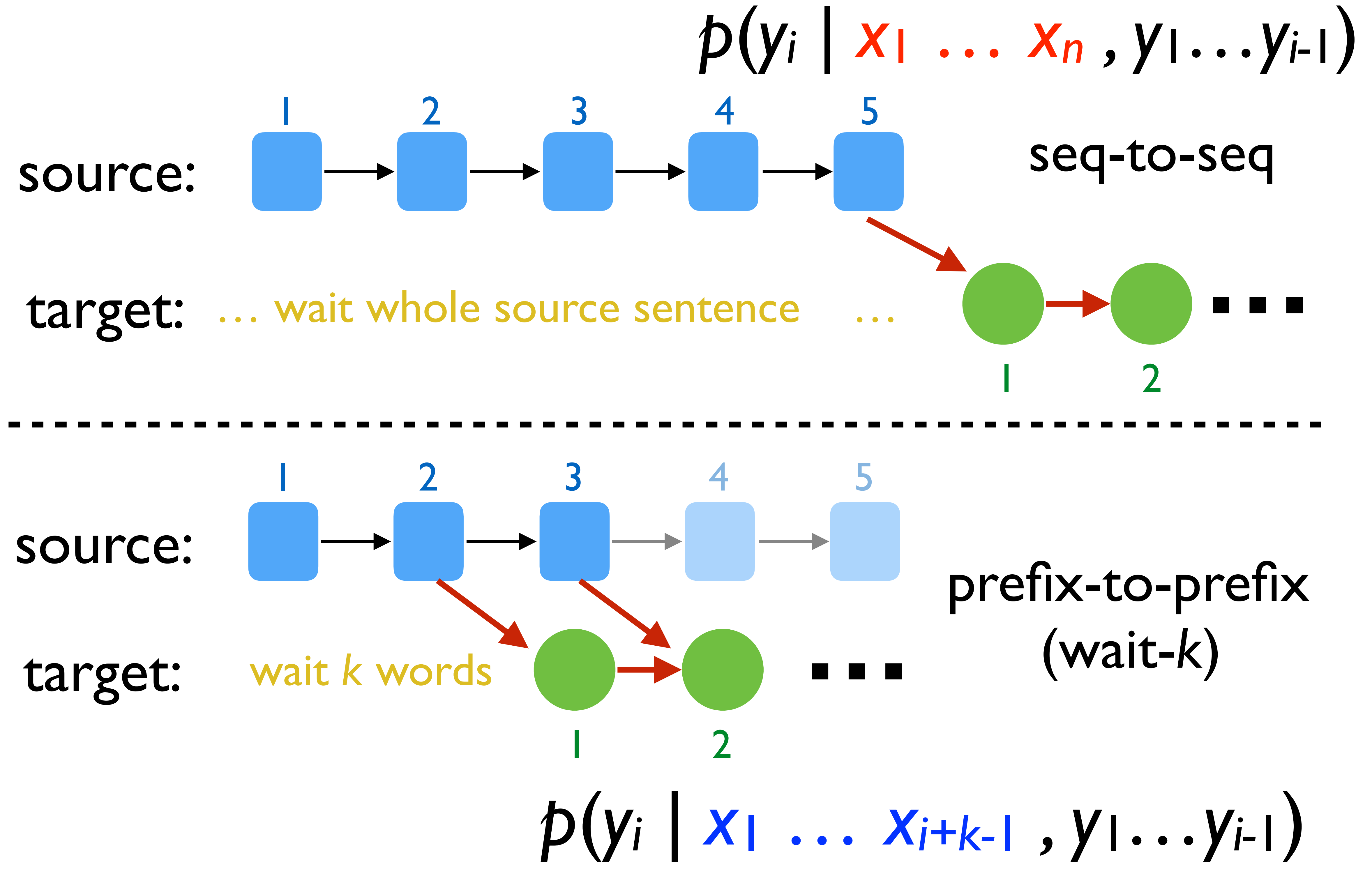


Bùshí	zǒngtǒng	zài	Mòsīkē
布什	总统	在	莫斯科
Bush	*President*	*in*	*Moscow*

wait 2 President Bush meets

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

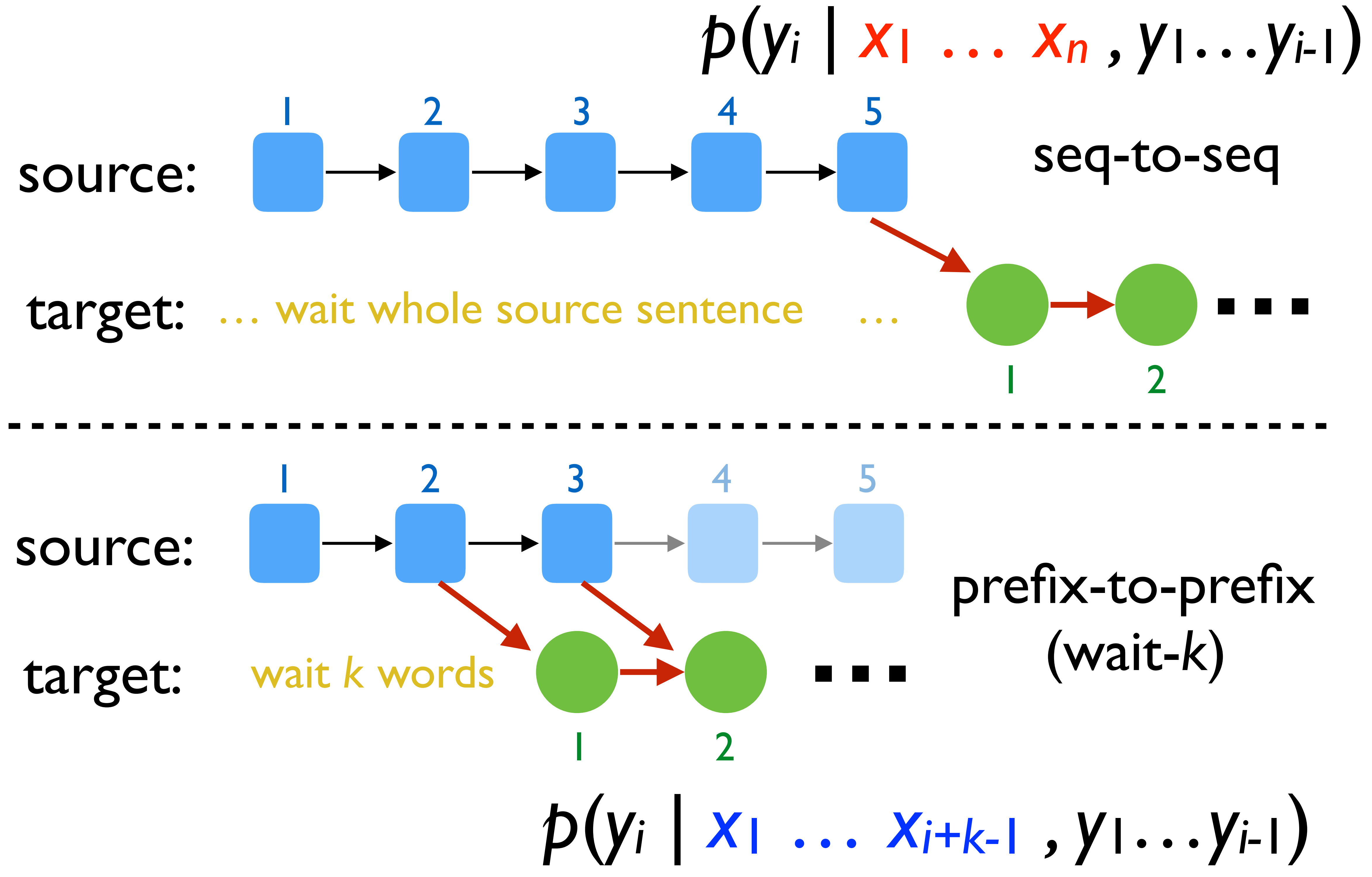


Bùshí	*zǒngtǒng*	*zài*	*Mòsīkē*	*yǔ*
布什	总统	在	莫斯科	与
Bush	*President*	*in*	*Moscow*	*with*

wait 2 President Bush meets with

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation



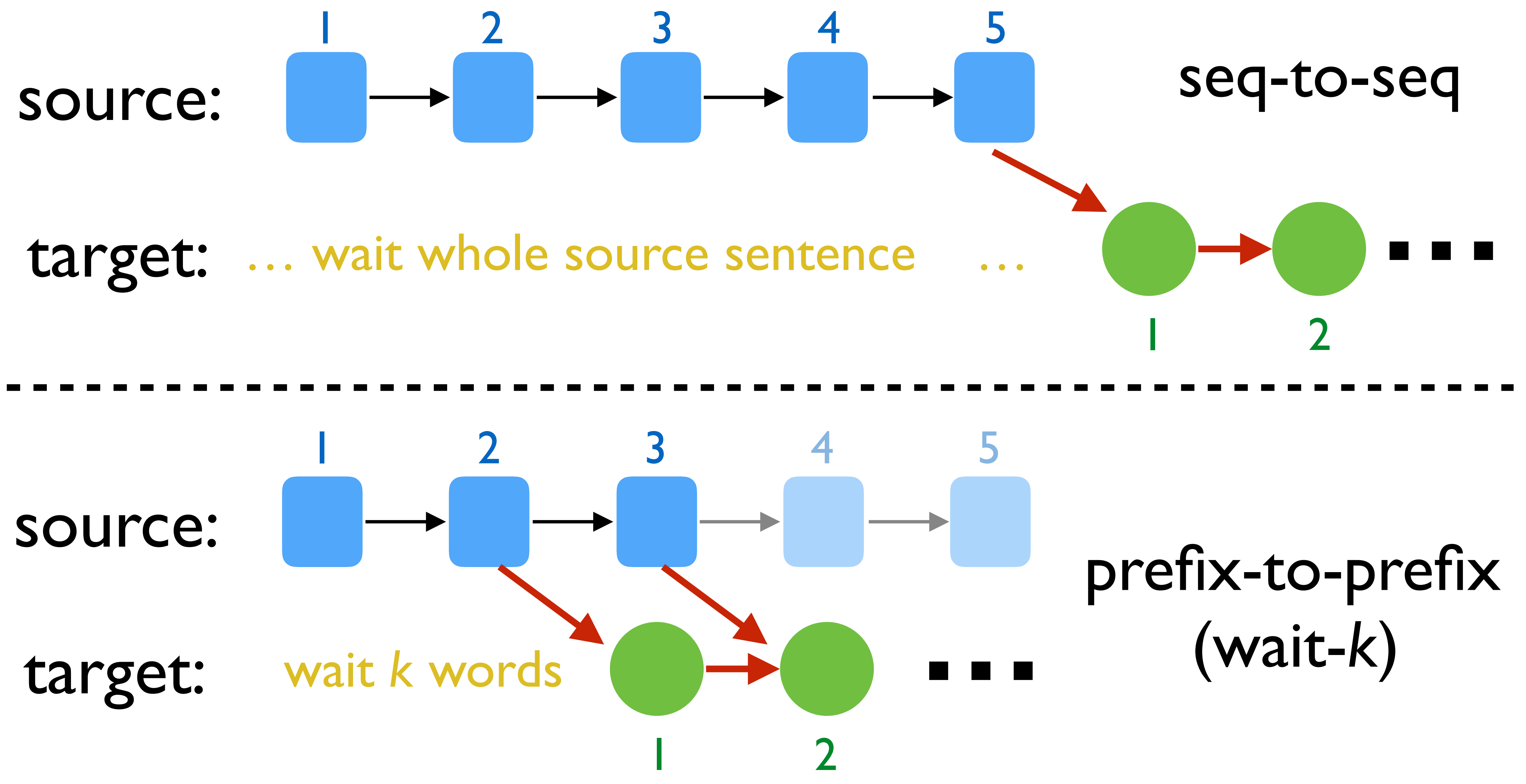
Bùshí	*zǒngtǒng*	*zài*	*Mòsīkē*	*yǔ*	*Éluósī*
布什	总统	在	莫斯科	与	俄罗斯
Bush	*President*	*in*	*Moscow*	*with*	*Russian*

wait 2 President Bush meets with Russian

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

$p(y_i \mid x_1 \ldots x_n, y_1 \ldots y_{i-1})$



$p(y_i \mid x_1 \ldots x_{i+k-1}, y_1 \ldots y_{i-1})$

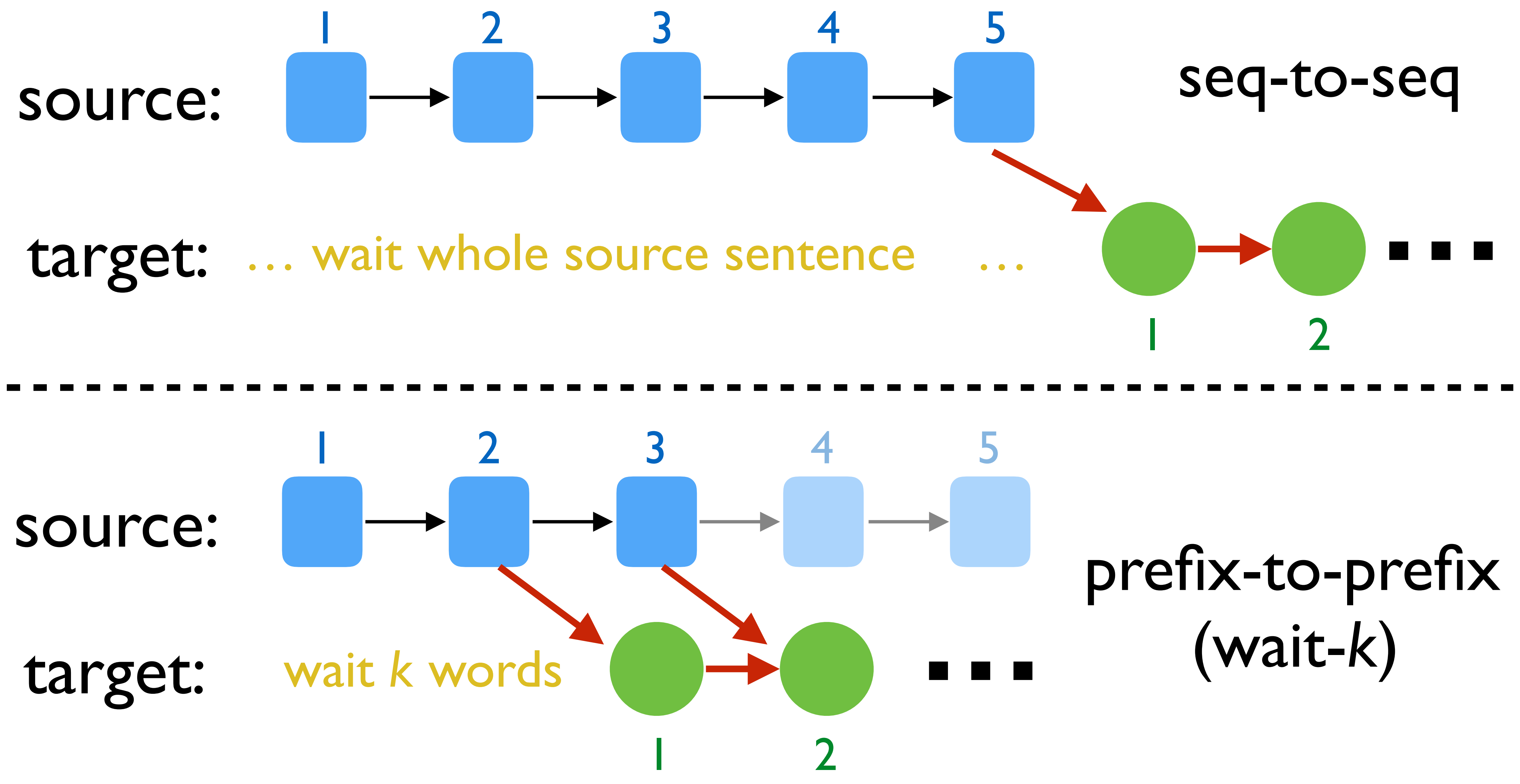
Bùshí	*zǒngtǒng*	*zài*	*Mòsīkē*	*yǔ*	*Éluósī*	*zǒngtǒng*
布什	总统	在	莫斯科	与	俄罗斯	总统
Bush	*President*	*in*	*Moscow*	*with*	*Russian*	*President*

wait 2 President Bush meets with Russian President

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

$p(y_i \mid x_1 \ldots x_n, y_1 \ldots y_{i-1})$



$p(y_i \mid x_1 \ldots x_{i+k-1}, y_1 \ldots y_{i-1})$

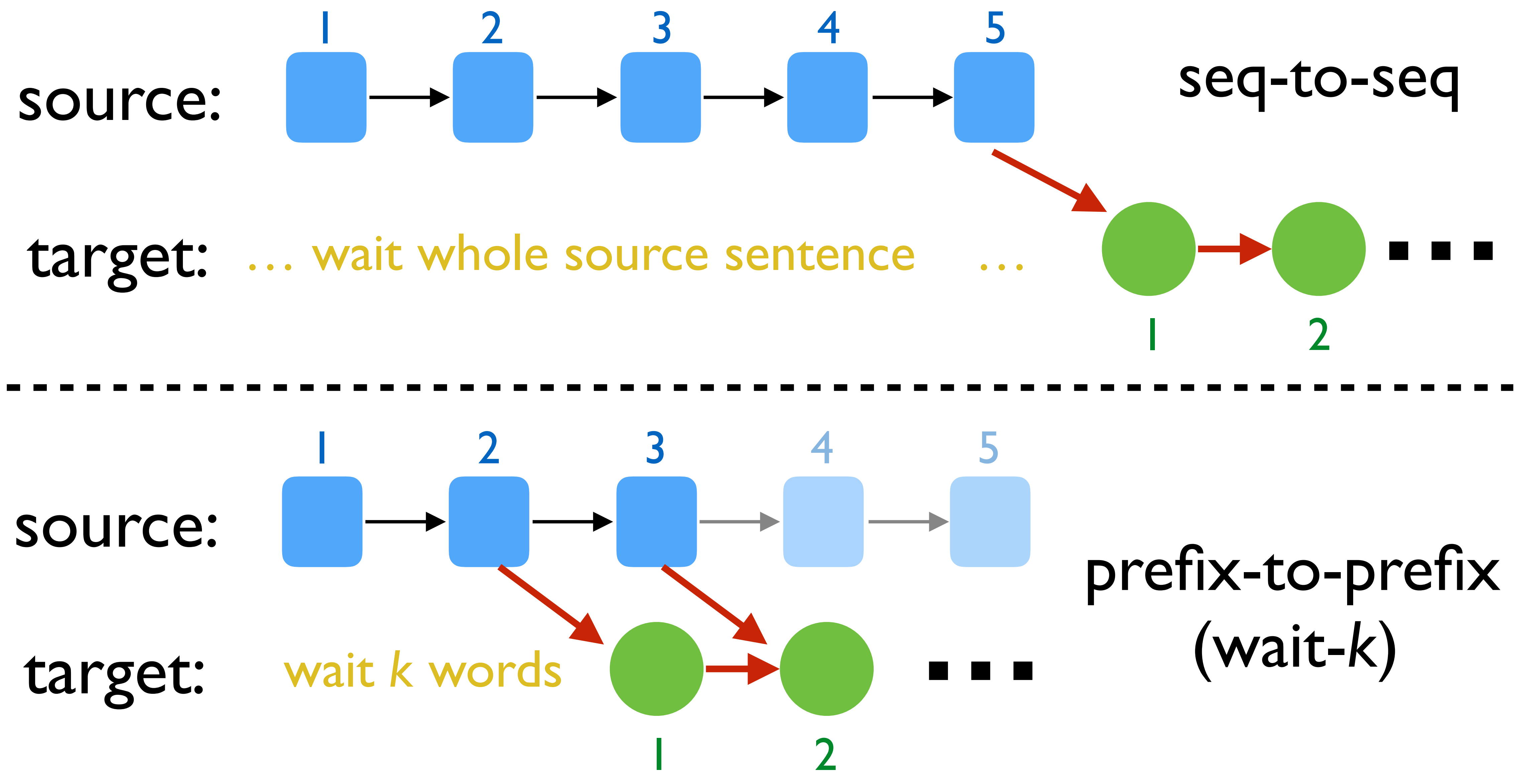
Bùshí	zǒngtǒng	zài	Mòsīkē	yǔ	Éluósī	zǒngtǒng	Pǔjīng
布什	总统	在	莫斯科	与	俄罗斯	总统	普京
Bush	President	in	Moscow	with	Russian	President	Putin

wait 2 President Bush meets with Russian President Putin

Our Solution: Prefix-to-Prefix

- standard seq-to-seq is only suitable for conventional full-sentence MT
- we propose prefix-to-prefix, tailed to simultaneous MT
  - special case: wait-*k* policy: translation is always *k* words behind source sentence
  - training in this way enables anticipation

$$p(y_i \mid x_1 \ldots x_n, y_1 \ldots y_{i-1})$$



$$p(y_i \mid x_1 \ldots x_{i+k-1}, y_1 \ldots y_{i-1})$$

Bùshí	zǒngtǒng	zài	Mòsīkē	yǔ	Éluósī	zǒngtǒng	Pǔjīng	huìwù
布什	总统	在	莫斯科	与	俄罗斯	总统	普京	会晤
Bush	President	in	Moscow	with	Russian	President	Putin	meet

wait 2 President Bush meets with Russian President Putin in Moscow

Research Demo

江泽民对法国总统的来华

jiang zemin expressed his appreciation

This is just our research demo. Our production system is better (shorter ASR latency).

Research Demo

Baidu Research This is just our research demo. Our production system is better (shorter ASR latency).

Research Demo

jiāng zémín	duì	fǎ guó	zǒng tǒng	de	lái huá	fǎng wèn	biǎo shì	gǎn xiè	
江 泽民	对	法国	总统	的	来华	访问	表示	感谢	。
jiang zemin	to	French	President	's	to-China	visit	express	gratitude	

jiang zemin expressed his appreciation for the visit by french president .

This is just our research demo. Our production system is better (shorter ASR latency).

Latency-Accuracy Tradeoff

Latency-Accuracy Tradeoff

Deployment Demo

This is live recording from the Baidu World Conference on Nov 1, 2018.

Deployment Demo

Baidu Research This is live recording from the Baidu World Conference on Nov 1, 2018.

Experimental Results (German=>English)

German source:

doch während man sich im kongress nicht auf ein vorgehen einigen kann , warten mehrere bundesstaaten nicht länger .

but while they self in congress not on one action agree can wait several states not longer

English translation (simultaneous, wait 3):

but , while congress <u>does</u> not agree on a course of action , several states no longer wait .

English translation (full-sentence baseline):

but , while congressional action can not be agreed , several states are no longer waiting .

Experimental Results (German=>English)

German source:

doch während man sich im kongress nicht auf ein vorgehen einigen kann , warten mehrere bundesstaaten nicht länger .

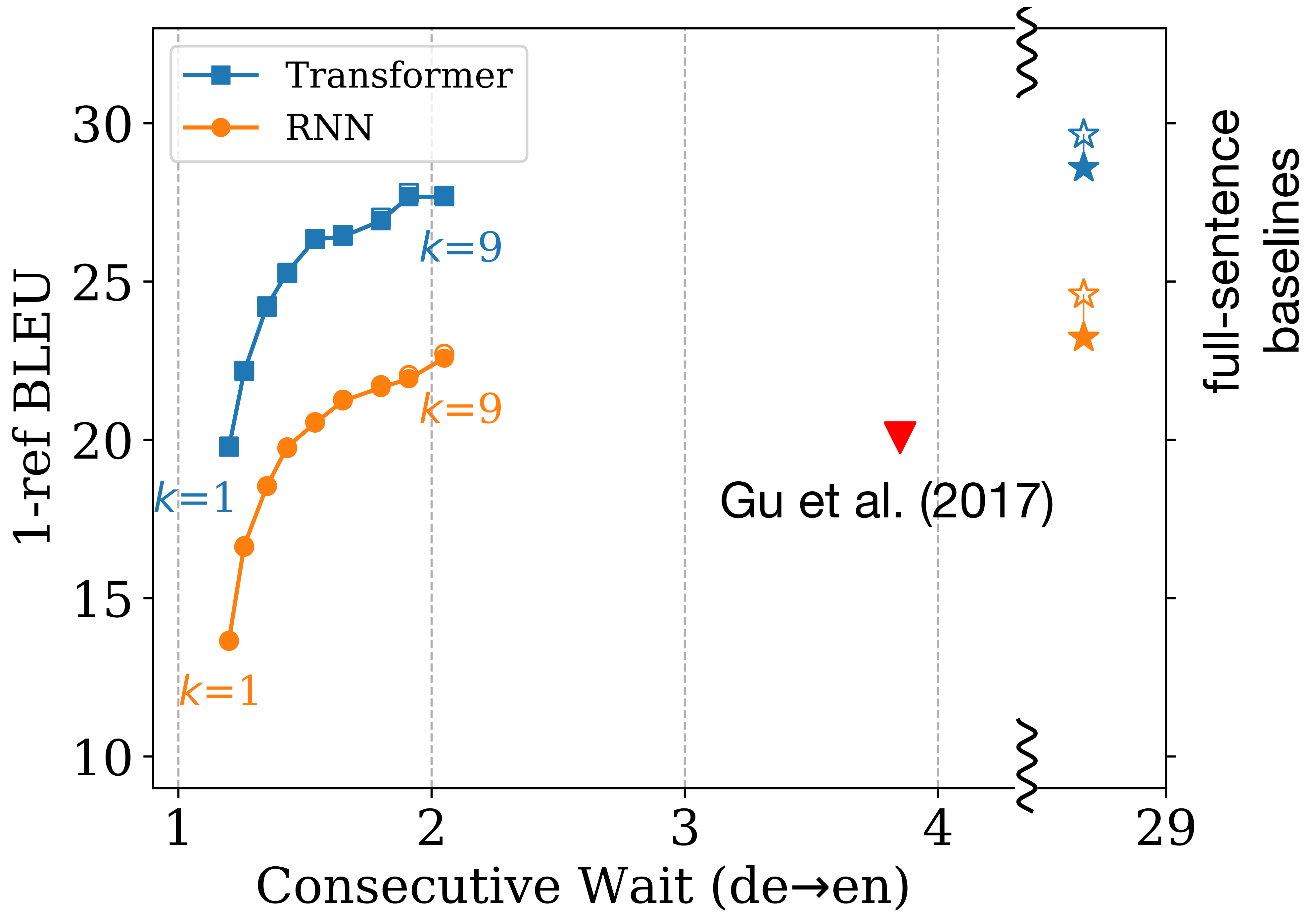
but while they self in congress not on one action agree can wait several states not longer

English translation (simultaneous, wait 3):

but , while congress does not agree on a course of action , several states no longer wait .

English translation (full-sentence baseline):

but , while congressional action can not be agreed , several states are no longer waiting .



Experimental Results (German=>English)

German source:

doch während man sich im kongress nicht auf ein vorgehen einigen kann , warten mehrere bundesstaaten nicht länger .

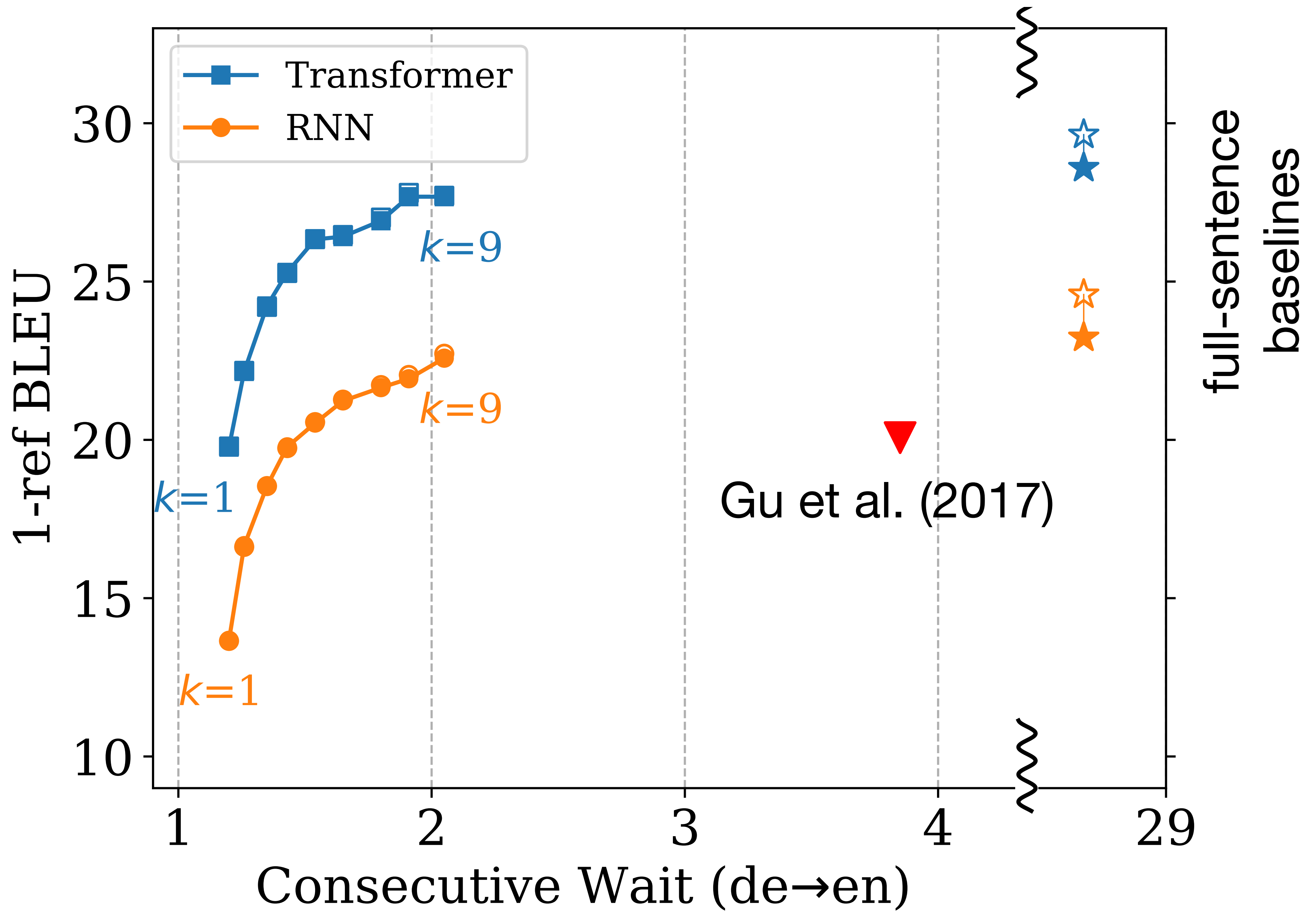
but while they self in congress not on one action agree can wait several states not longer

English translation (simultaneous, wait 3):

but , while congress <u>does</u> not agree on a course of action , several states no longer wait .

English translation (full-sentence baseline):

but , while congressional action can not be agreed , several states are no longer waiting .



ich bin mit dem Zug nach Ulm **gefahren**

I am with the train to Ulm **traveled**

wait 8 words | I traveled to Ulm by train

full-sentence baseline: CW = 8

wait 2 | I | wait 6 words | traveled to Ulm by train

Gu et al. (2017): CW = (2+6)/2 = 4

wait 4 | I | 1 | took | 1 | a | 1 | train | 1 | to Ulm

our wait 4 model: CW = (4+1+1+1+1)/5 = 1.6

Summary of Innovations and Impact

- first simultaneous translation approach with integrated anticipation
  - inspired by human simultaneous interpreters who routinely anticipate
- first simultaneous translation approach with arbitrary controllable latency
  - previous RL-based approaches can encourage but can’t enforce latency limit
- very easy to train and scalable — minor changes to any neural MT codebase
- prefix-to-prefix is very general; can be used in other tasks with simultaneity

Summary of Innovations and Impact

- first simultaneous translation approach with integrated anticipation
  - inspired by human simultaneous interpreters who routinely anticipate
- first simultaneous translation approach with arbitrary controllable latency
  - previous RL-based approaches can encourage but can’t enforce latency limit
- very easy to train and scalable — minor changes to any neural MT codebase
- prefix-to-prefix is very general; can be used in other tasks with simultaneity

Media coverage:

Next: Integrate Incremental Predictive Parsing

- how to be smarter about when to wait and when to translate?

mandatory reordering (i.e., wait):

xíjìnpíng	yú		nián	zài	běijīng	dāngxuǎn
习近平	于	2012	年	在	北京	当选
Xi Jiping	in	2012	yr	in	Beijing	elected

reference translation

"Xi Jinping was elected in Beijing in 2012"

optional reordering:

guānyú	kèlíndùnzhǔyì		méiyǒu	zhǔnquè	de	dìngyì
关于	克林顿主义	，	没有	准确	的	定义
about	Clintonism		no	accurate		def.

"There is no accurate definition of Clintonism."

Next: Integrate Incremental Predictive Parsing

- how to be smarter about when to wait and when to translate?

mandatory reordering (i.e., wait):

optional reordering:

xíjìnpíng yú nián zài běijīng dāngxuǎn

习近平 于 2012 年 在 北京 当选

Xi Jiping in 2012 yr in Beijing elected

guānyú kèlíndùnzhǔyì méiyǒu zhǔnquè de dìngyì

关于 克林顿主义 , 没有 准确 的 定义

about Clintonism no accurate def.

reference translation

"Xi Jinping was elected in Beijing in 2012"

"There is no accurate definition of Clintonism."

ideal simultaneous

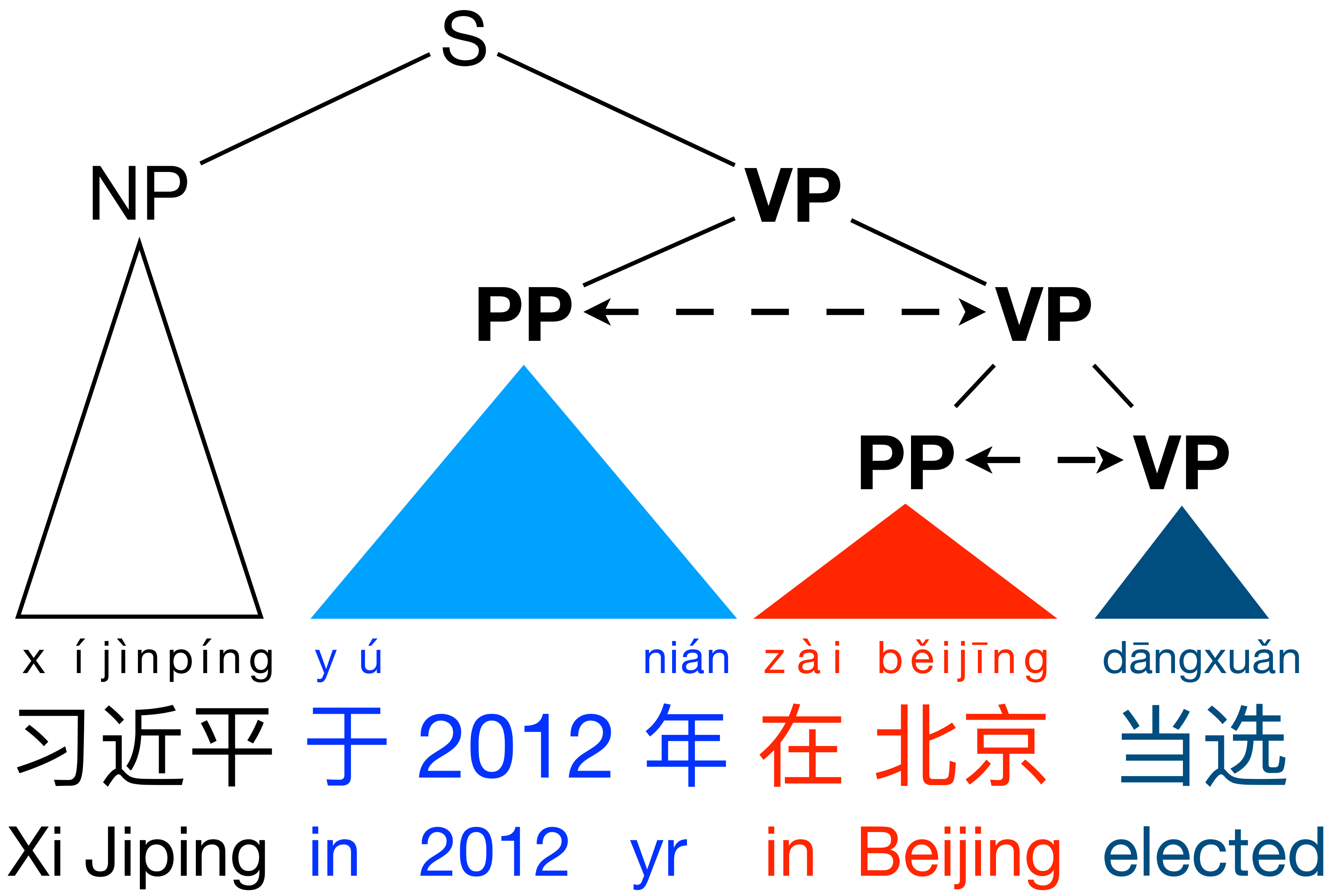
Xi Jinping ….. was elected…

About Clintonism, there is no accurate definition.

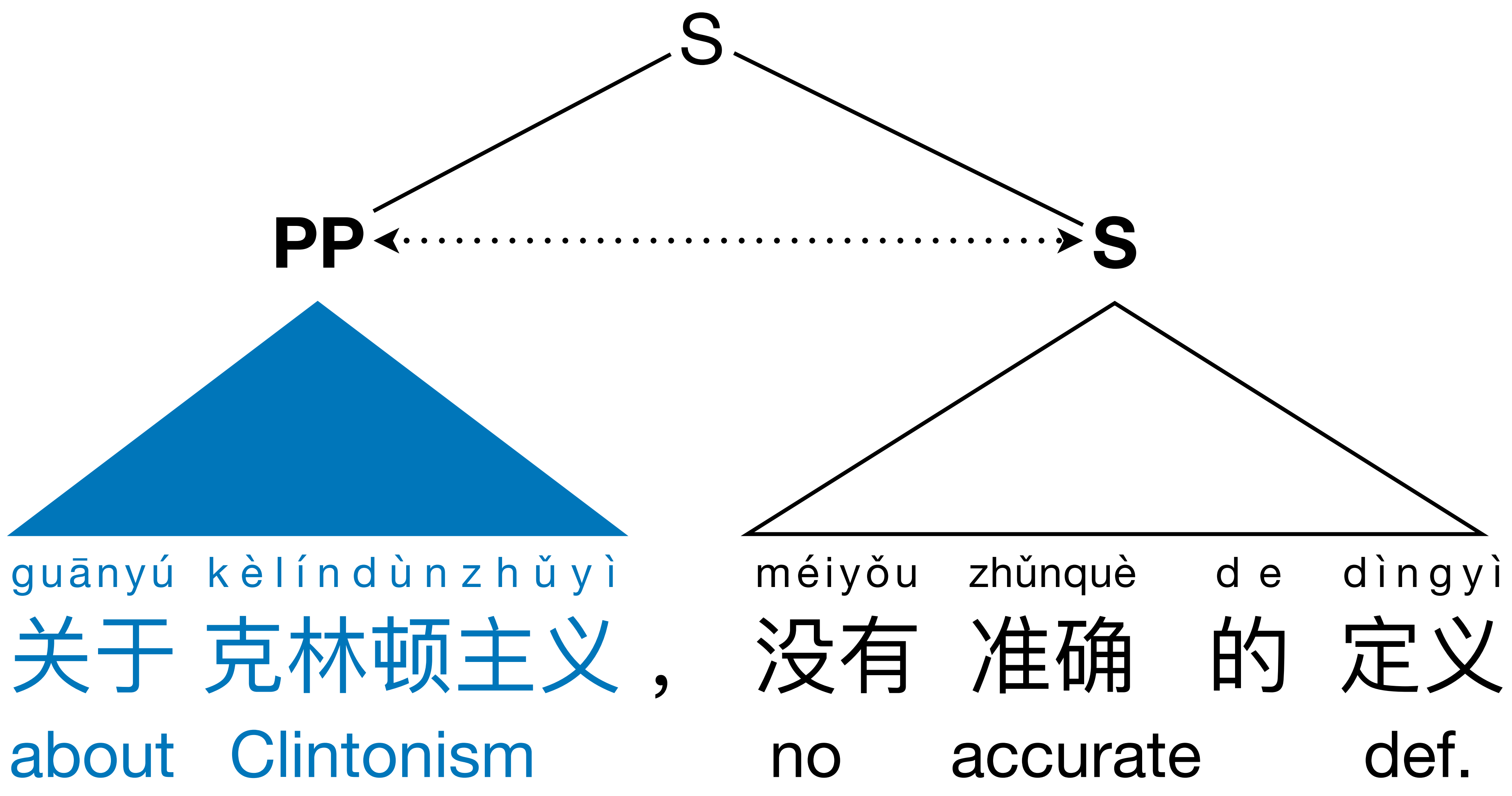
Next: Integrate Incremental Predictive Parsing

- how to be smarter about when to wait and when to translate?

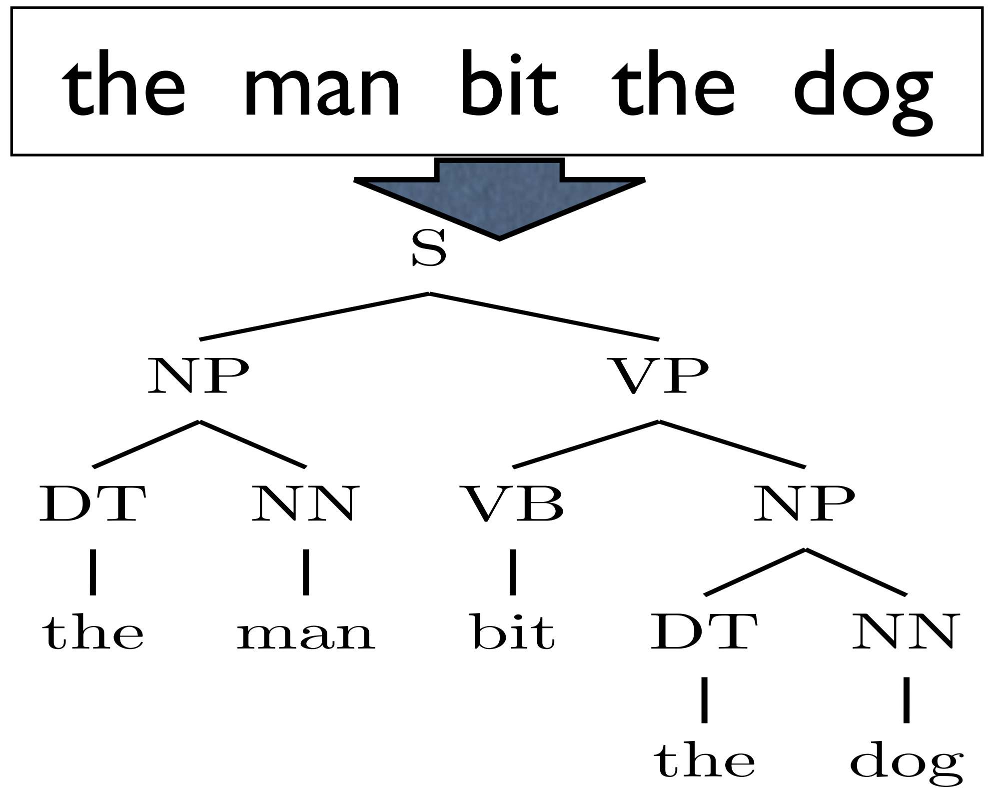
mandatory reordering (i.e., wait):
(Chinese) PP VP => (English) VP PP



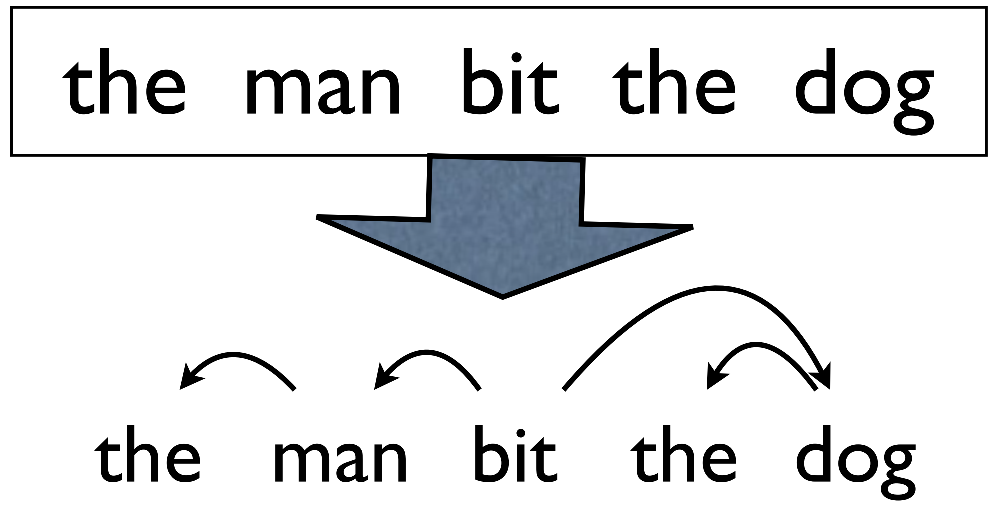
optional reordering:
(Chinese) PP S => (English) PP S or S PP



reference translation	"Xi Jinping was elected in Beijing in 2012"	"There is no accurate definition of Clintonism."
ideal simultaneous	Xi Jinping ….. was elected…	About Clintonism, there is no accurate definition.



constituency parsing

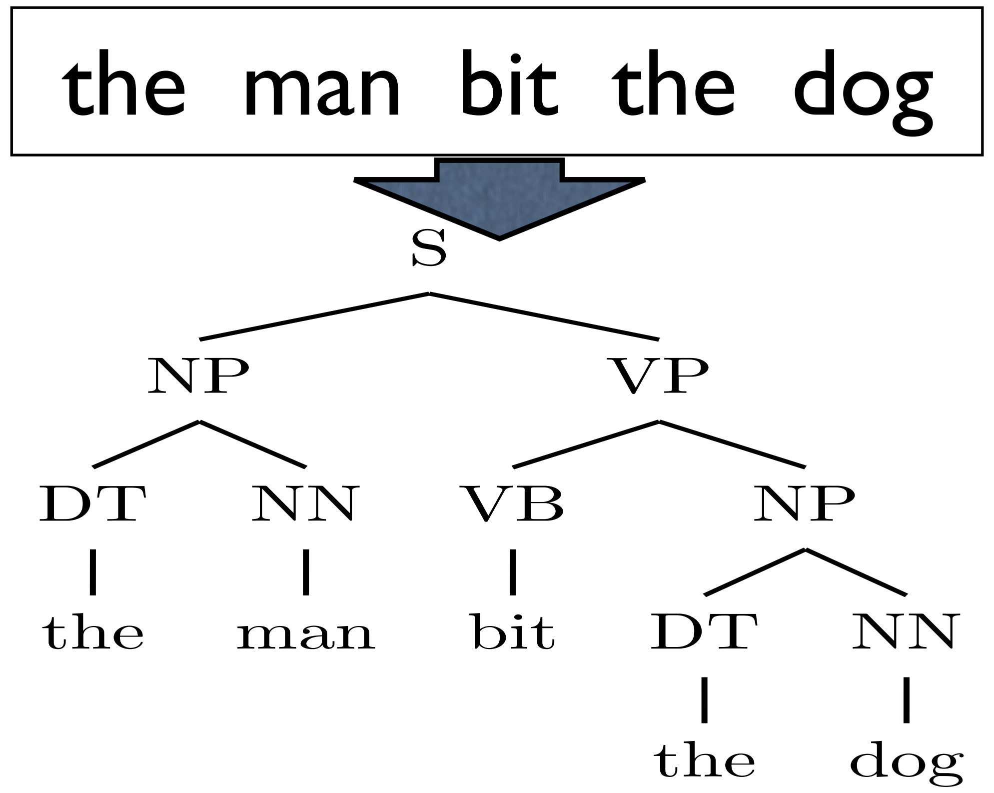


dependency parsing

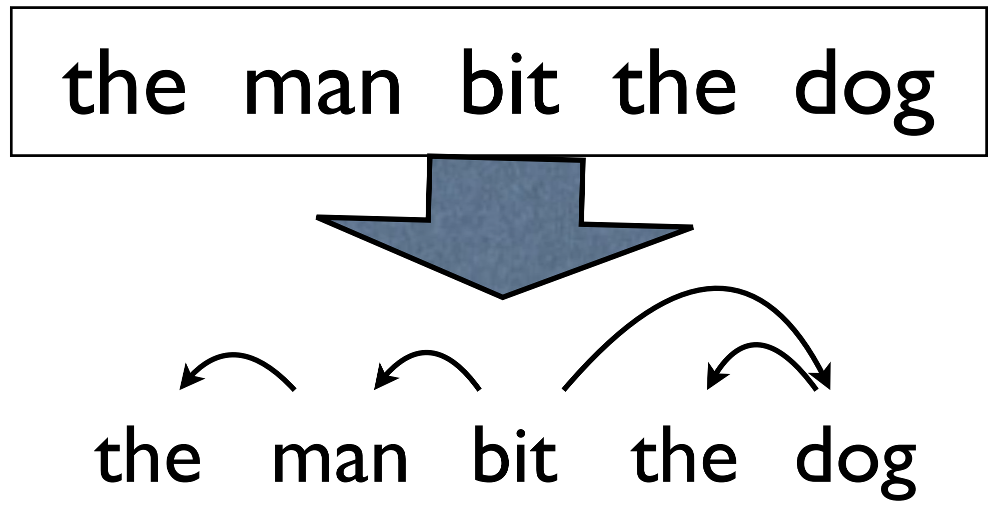
Part II: Linear-Time Incremental Parsing

(Huang & Sagae, ACL 2010*; Goldberg, Zhao, Huang, ACL 2013;
Zhao, Cross, Huang, EMNLP 2013; Mi & Huang, ACL 2015;
Cross & Huang, ACL 2016; Cross & Huang, EMNLP 2016**
Hong and Huang, ACL 2018)

* *best paper nominee* ** *best paper honorable mention*



constituency parsing



dependency parsing

Part II: Linear-Time Incremental Parsing

(Huang & Sagae, ACL 2010*; Goldberg, Zhao, Huang, ACL 2013;
Zhao, Cross, Huang, EMNLP 2013; Mi & Huang, ACL 2015;
Cross & Huang, ACL 2016; Cross & Huang, EMNLP 2016**
Hong and Huang, ACL 2018)

* *best paper nominee* ** *best paper honorable mention*

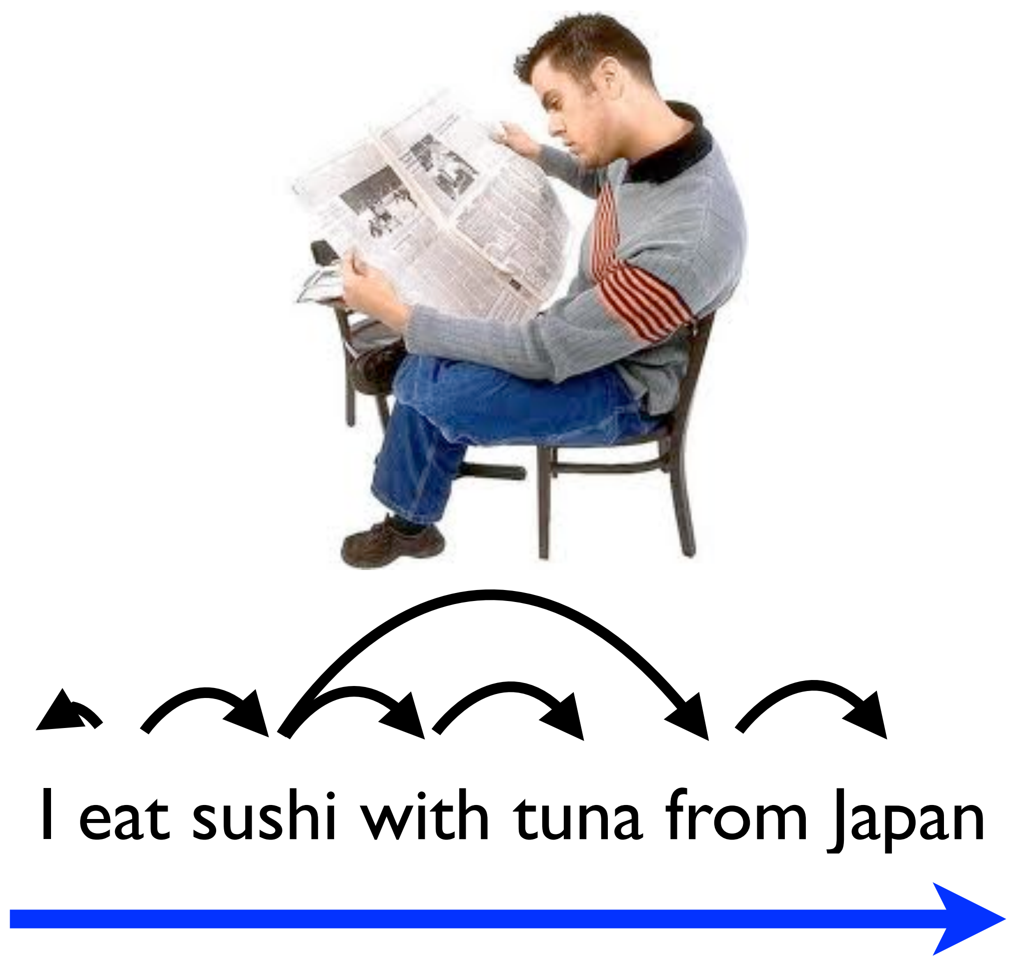
Motivations for Incremental Parsing

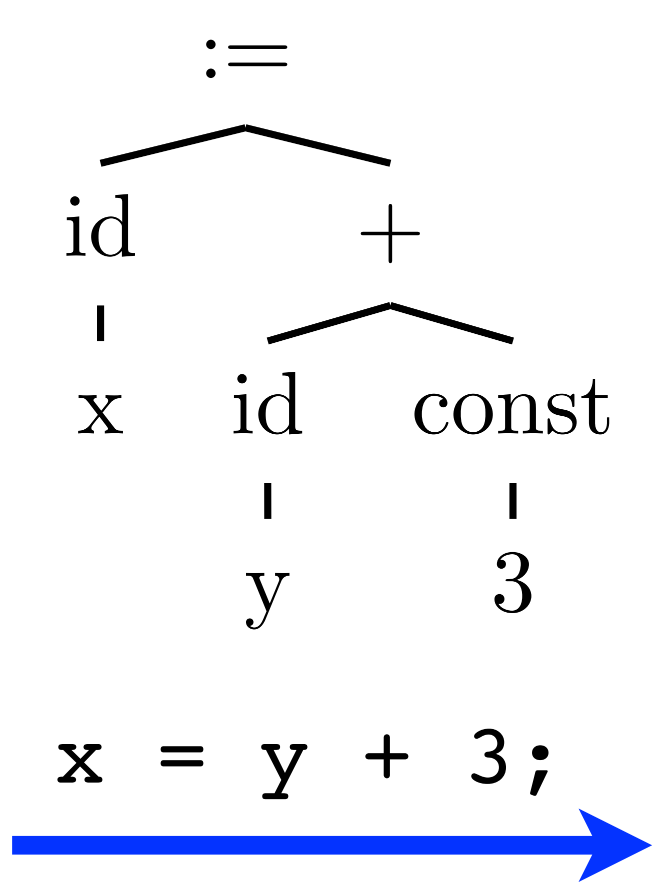
- simultaneous translation
- auto completion (search suggestions)
- question answering
- dialog
- speech recognition
- input method editor
- …

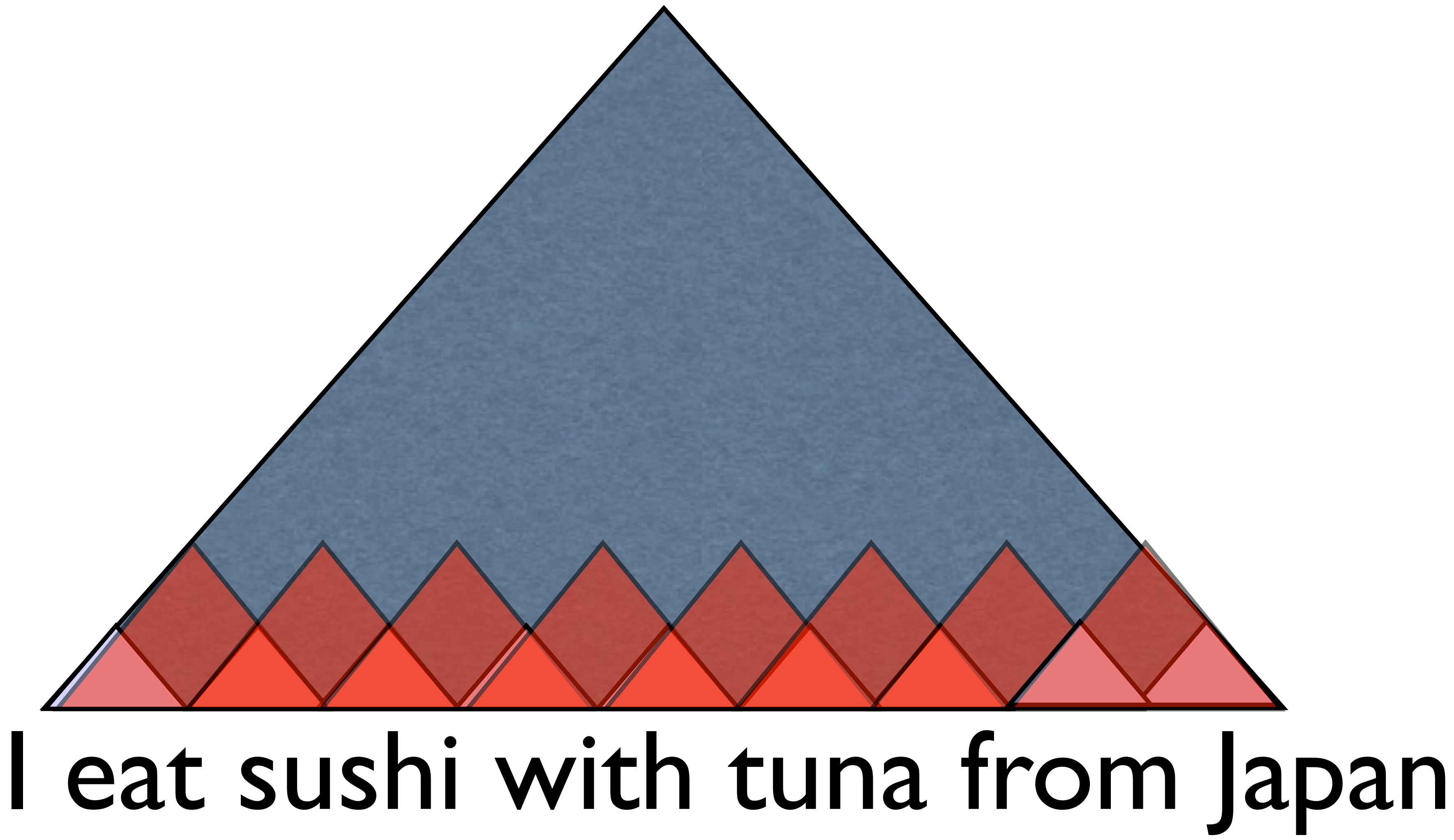
siri is so

siri is so **dumb**
siri is so **bad**
siri is so **useless**
siri is so **slow**
siri is so **rude**

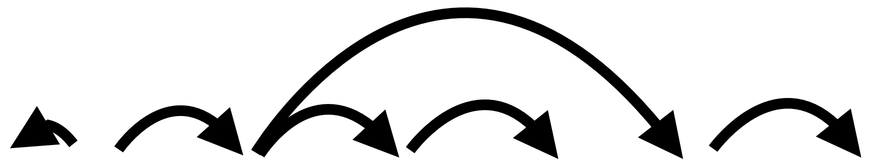
Human Parsing vs. Compilers vs. NL Parsing





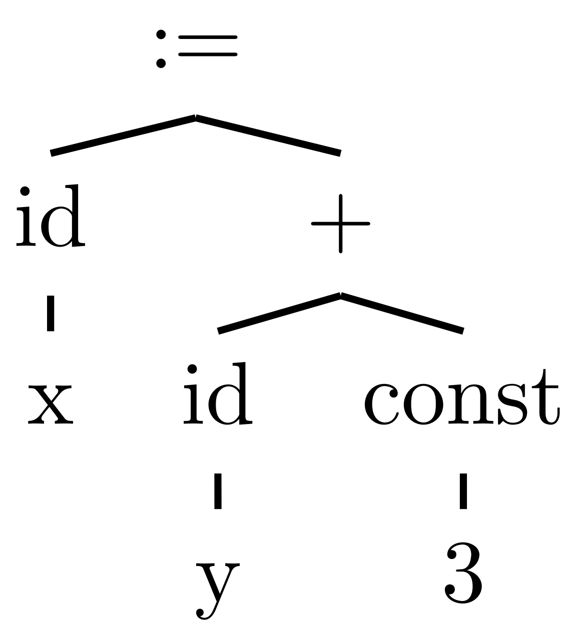


Human Parsing vs. Compilers vs. NL Parsing



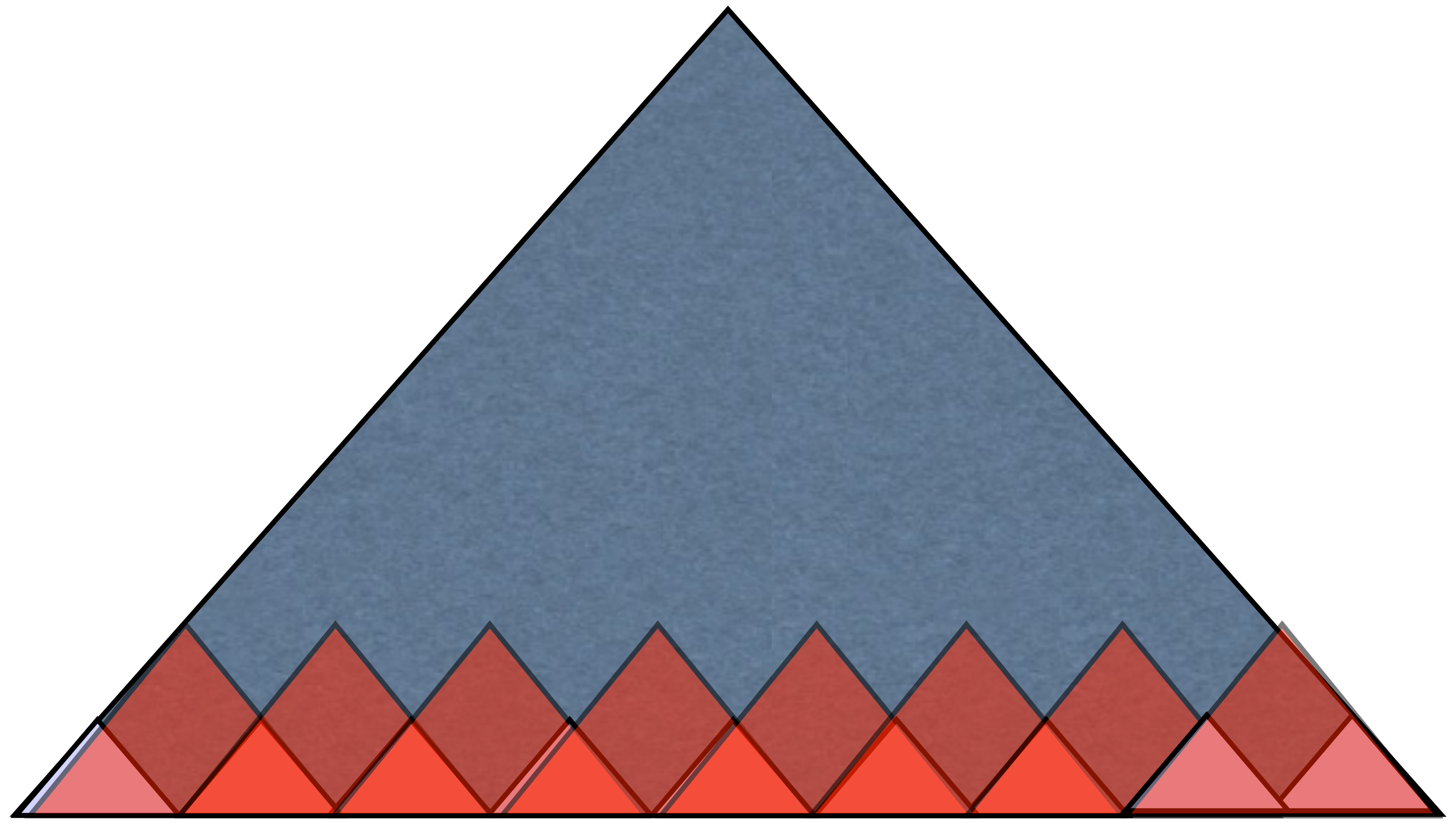
I eat sushi with tuna from Japan

$O(n)$




```
x = y + 3;
```

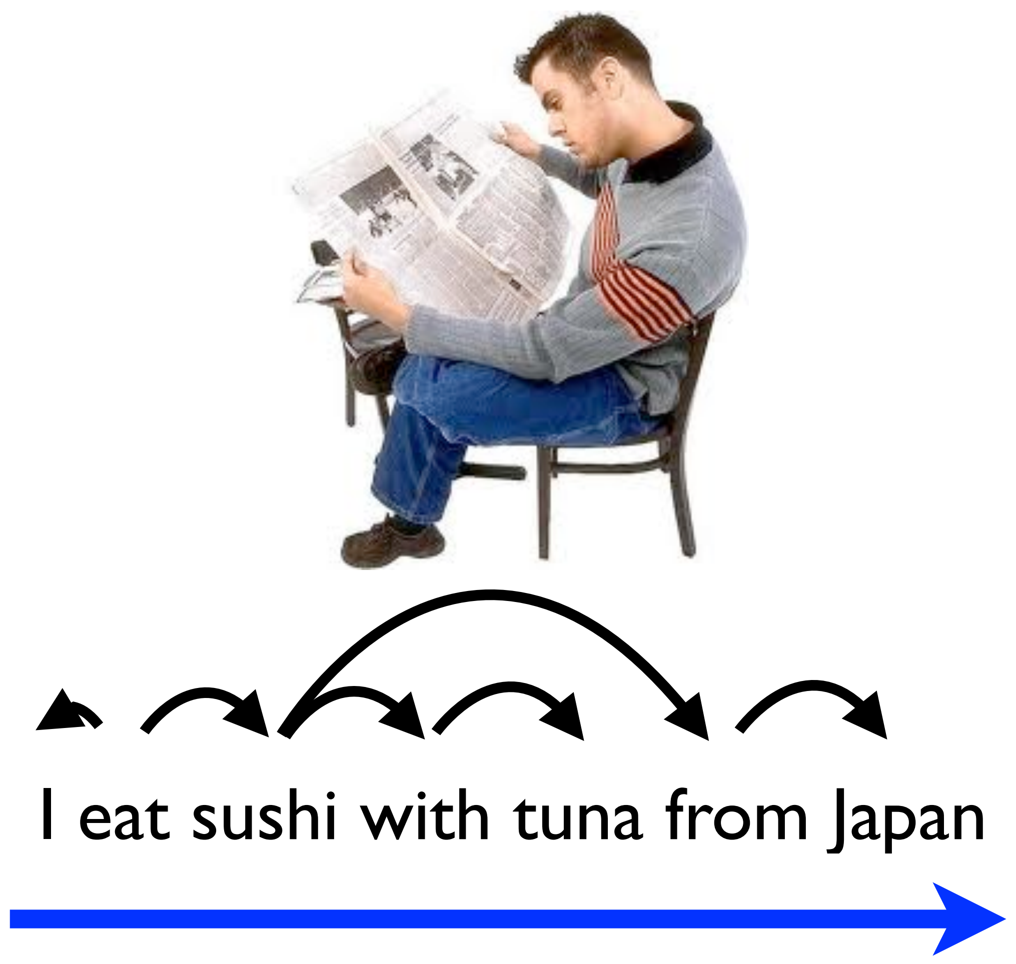
$O(n)$



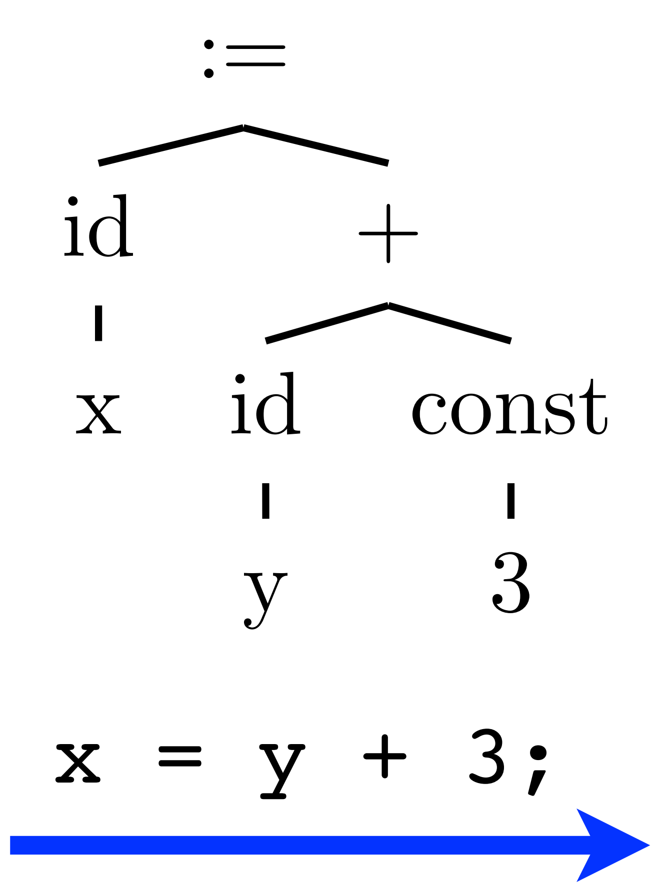
I eat sushi with tuna from Japan

$O(n^3)$

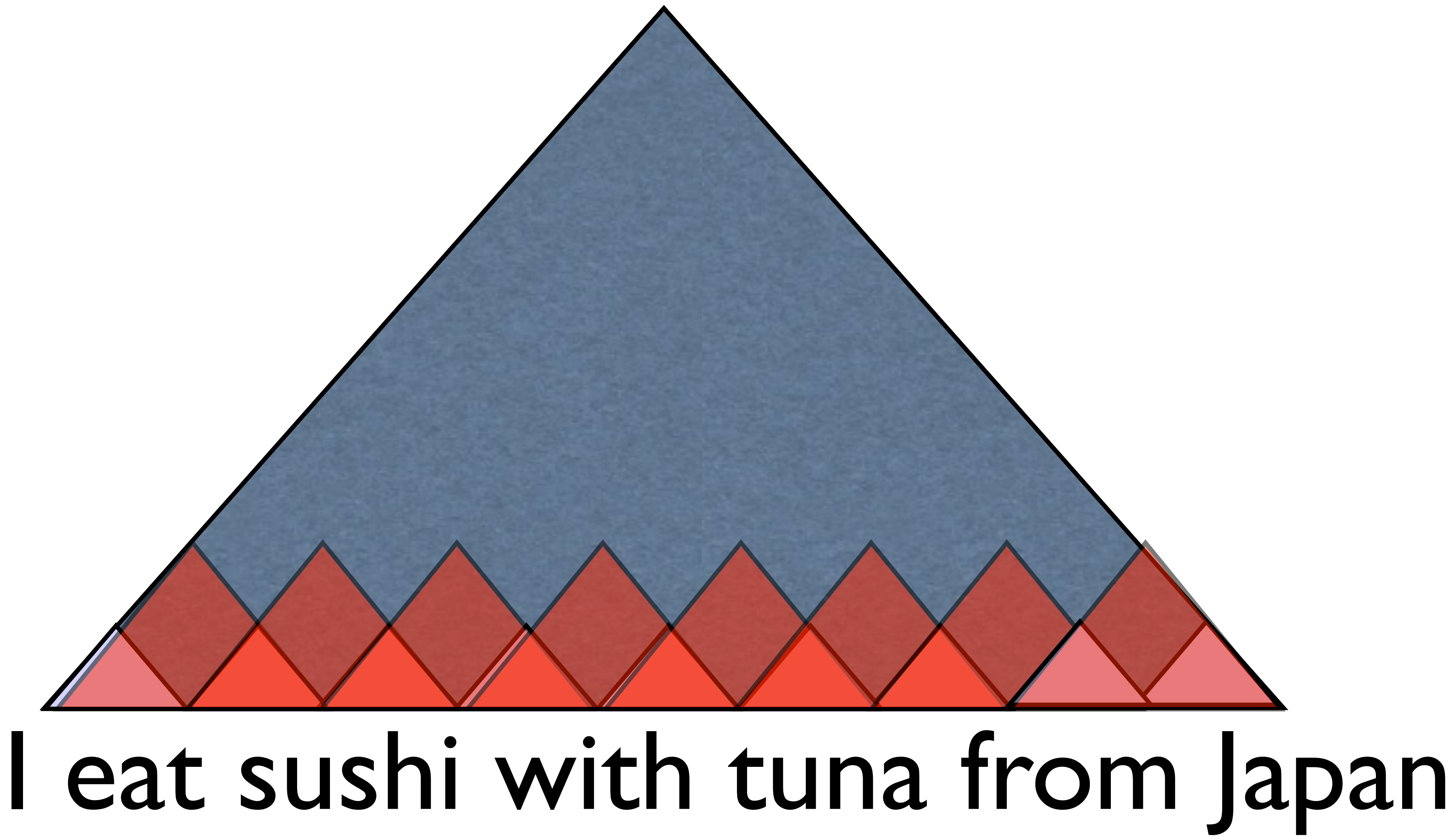
Human Parsing vs. Compilers vs. NL Parsing



$O(n)$



$O(n)$

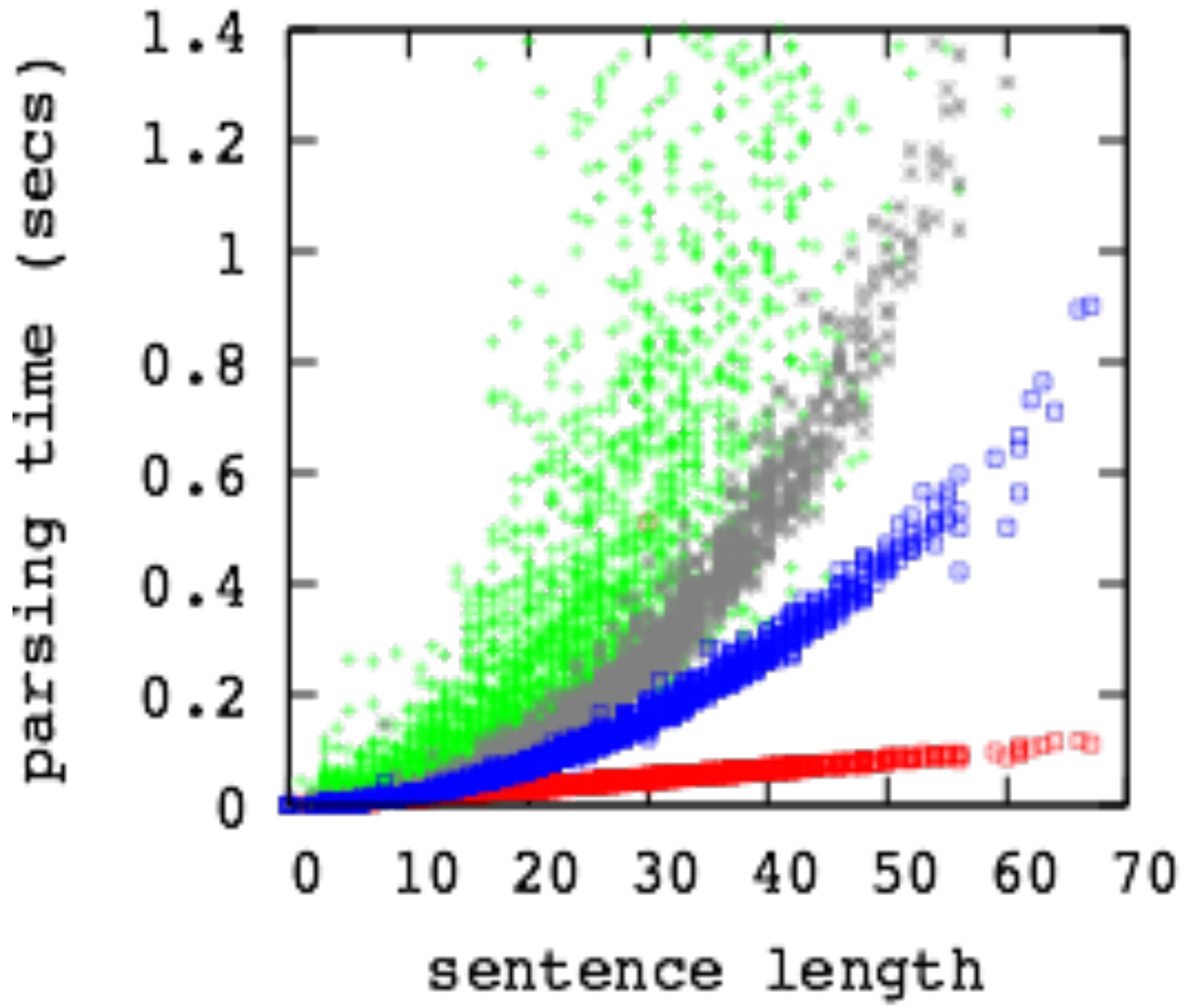


$O(n^3)$

- can we design NL parsing algorithms that is both fast and accurate, inspired by human sentence processing and compilers?
- our idea: generalize PL parsing (LR algorithm) to NL parsing, but keep it $O(n)$
- challenge: how to deal with ambiguity ***explosion*** in NL?
- solution: linear-time dynamic programming — both fast and accurate!

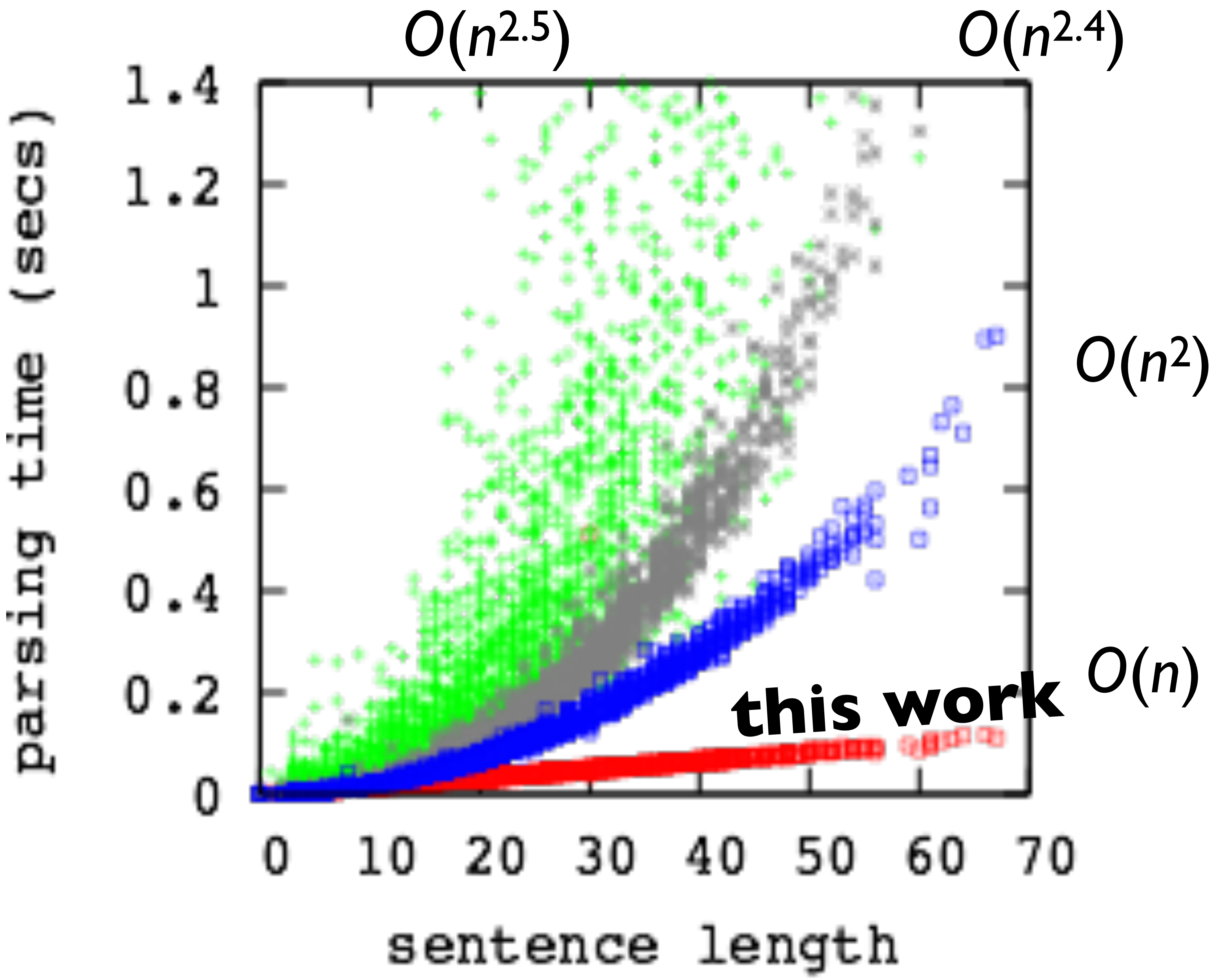
Solution: linear-time, DP, and accurate!

- very fast linear-time dynamic programming parser
- explores *exponentially* many trees (and outputs forest)
- accurate parsing accuracy on English & Chinese



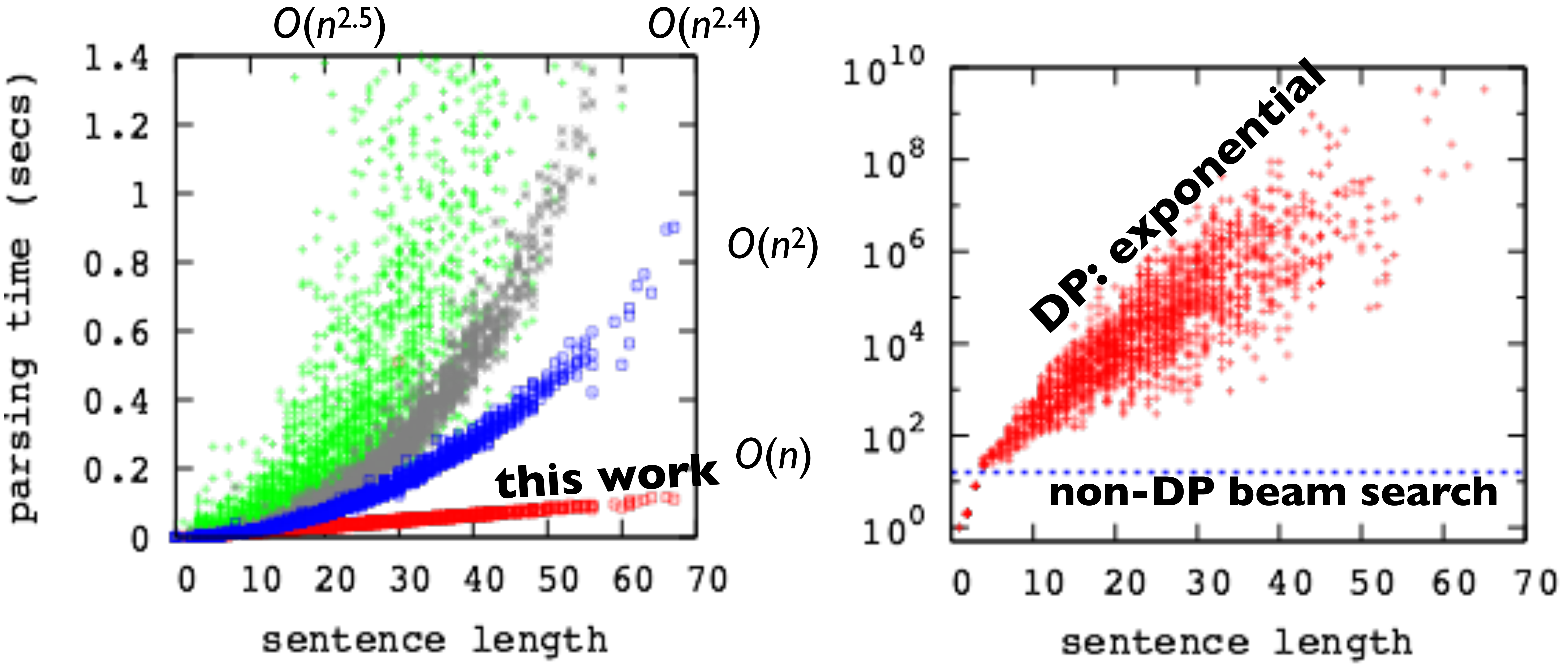
Solution: linear-time, DP, and accurate!

- very fast linear-time dynamic programming parser
- explores *exponentially* many trees (and outputs forest)
- accurate parsing accuracy on English & Chinese



Solution: linear-time, DP, and accurate!

- very fast linear-time dynamic programming parser
- explores *exponentially* many trees (and outputs forest)
- accurate parsing accuracy on English & Chinese



Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

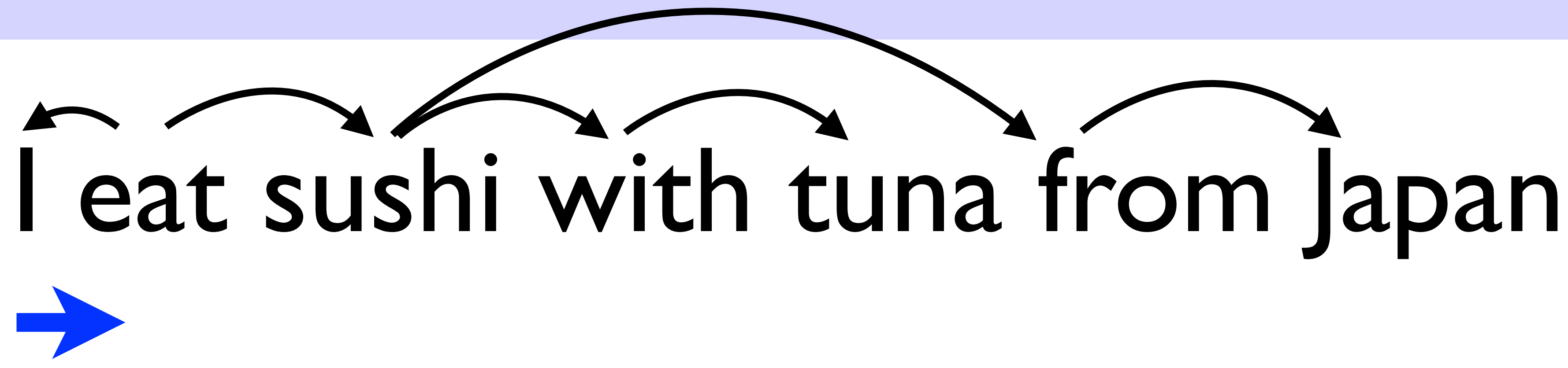
action stack queue

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

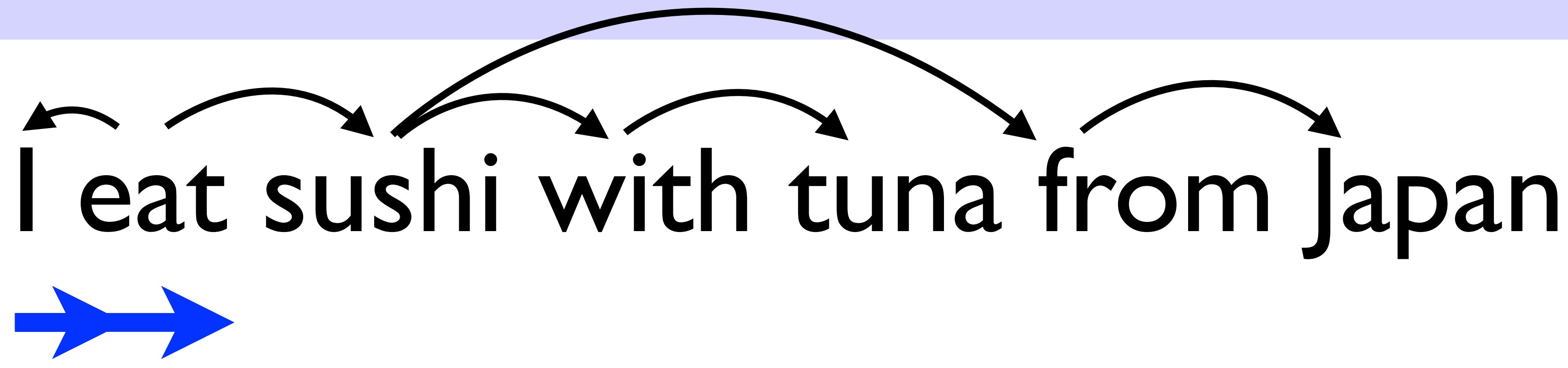
	action	stack	queue
0	-		I eat sushi ...

Incremental Parsing (Shift-Reduce)



	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...

Incremental Parsing (Shift-Reduce)



	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

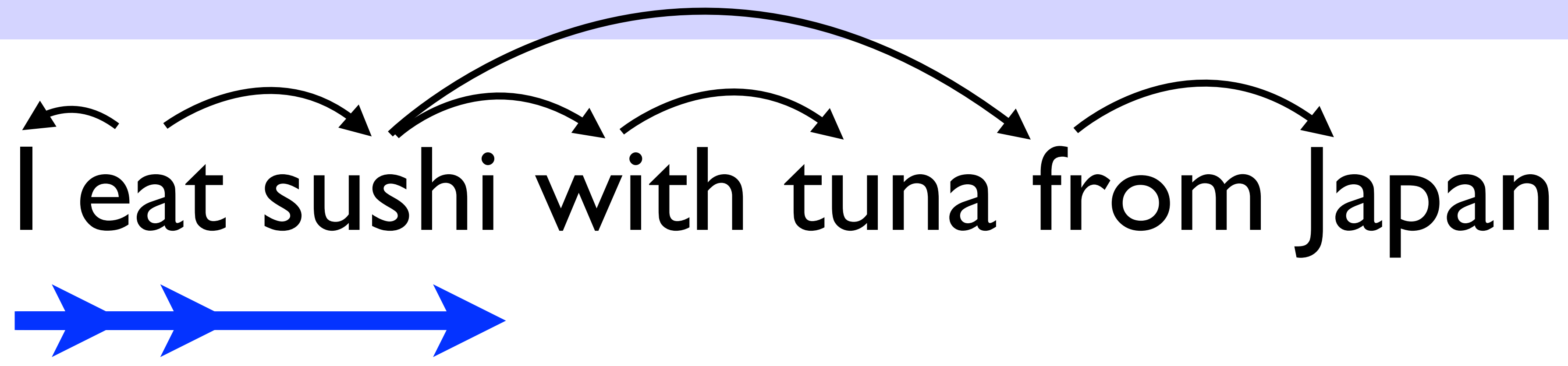
	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...
3	*l*-reduce	eat I	sushi with tuna ...

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...
3	*l*-reduce	eat I	sushi with tuna ...
4	shift	eat sushi I	with tuna from ...

Incremental Parsing (Shift-Reduce)



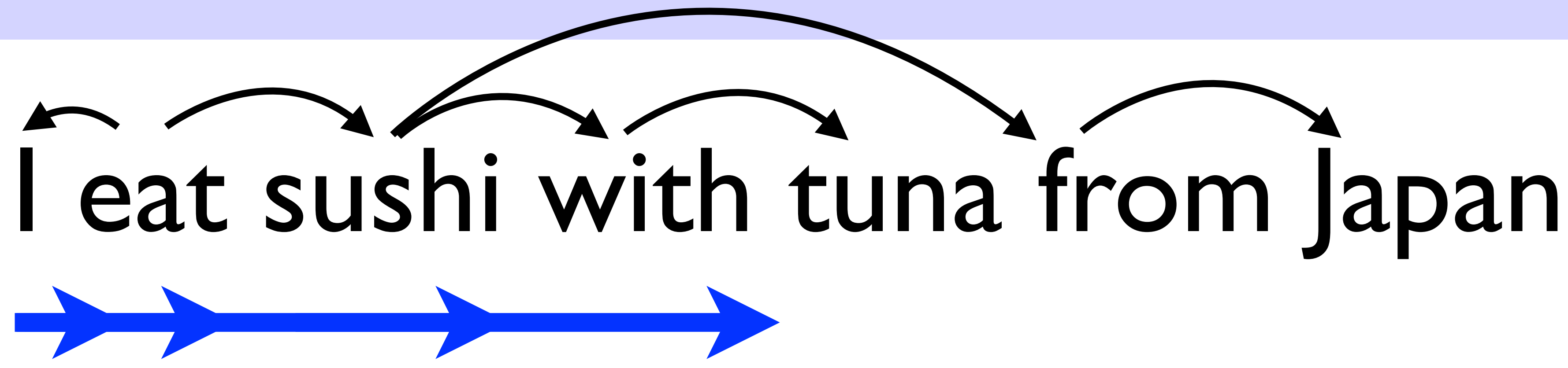
	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...
3	*l*-reduce	eat I	sushi with tuna ...
4	shift	eat sushi I	with tuna from ...
5a	*r*-reduce	eat I sushi	with tuna from ...

Incremental Parsing (Shift-Reduce)

I eat sushi with tuna from Japan

	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...
3	*l*-reduce	eat I	sushi with tuna ...
4	shift	eat sushi I	with tuna from ...
5a	*r*-reduce	eat I sushi	with tuna from ...
5b	shift	eat sushi with I	tuna from Japan ...

Incremental Parsing (Shift-Reduce)

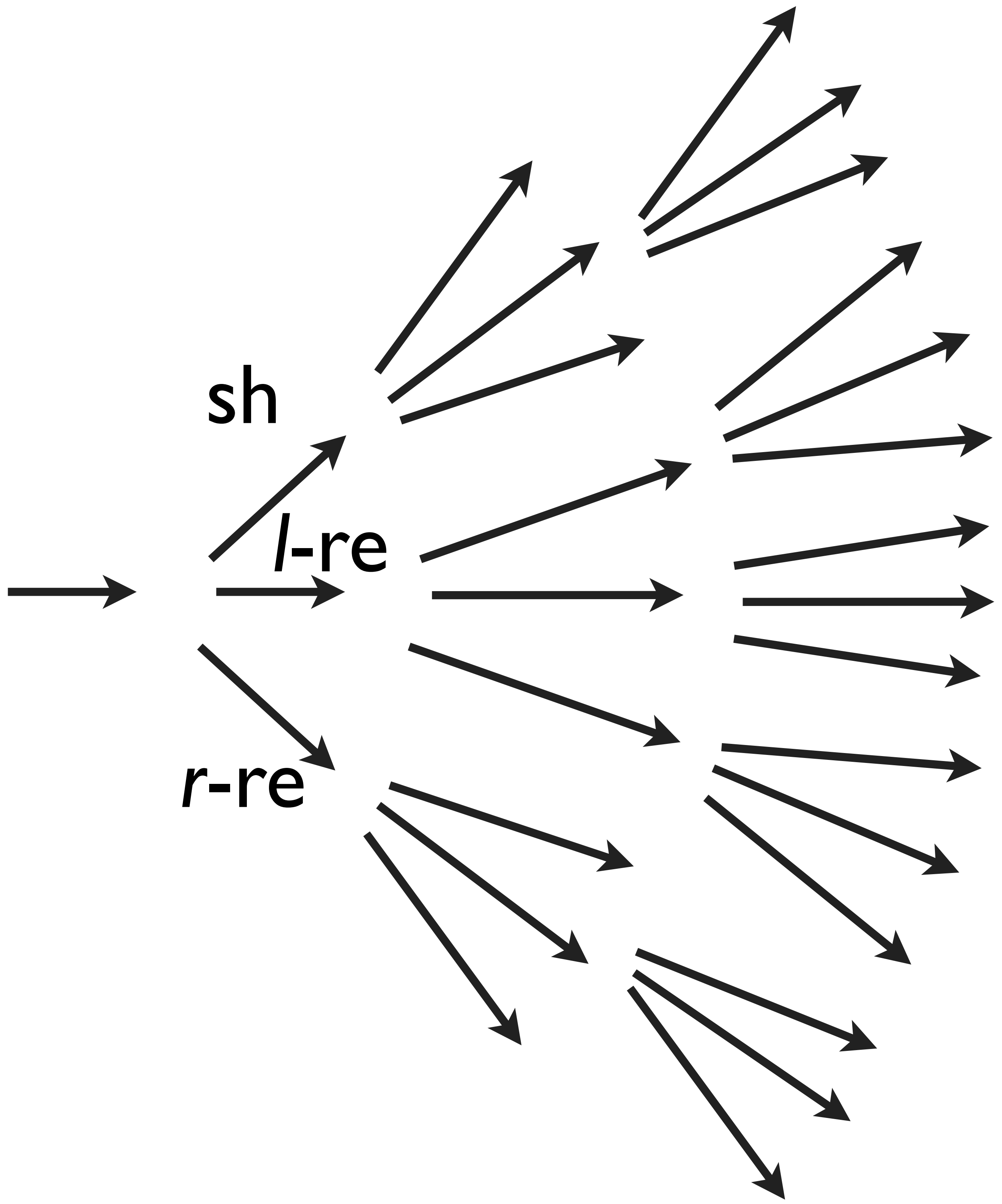


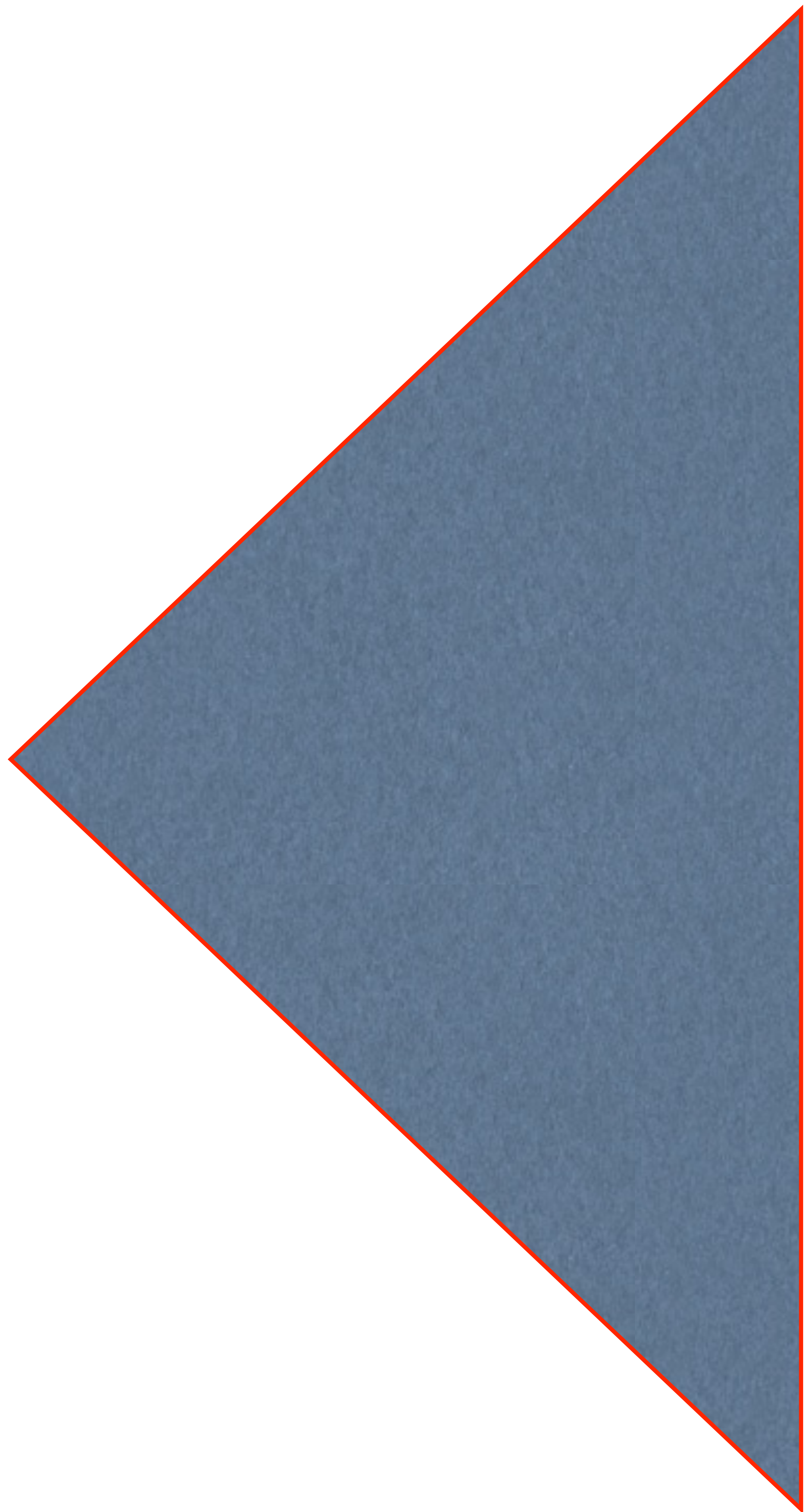
	action	stack	queue
0	-		I eat sushi ...
1	shift	I	eat sushi with ...
2	shift	I eat	sushi with tuna ...
3	*l*-reduce	eat	sushi with tuna ...
4	shift	eat sushi	with tuna from ...
5a	*r*-reduce	eat sushi	with tuna from ...
5b	shift	eat sushi with	tuna from Japan ...

shift-reduce conflict

Greedy Search

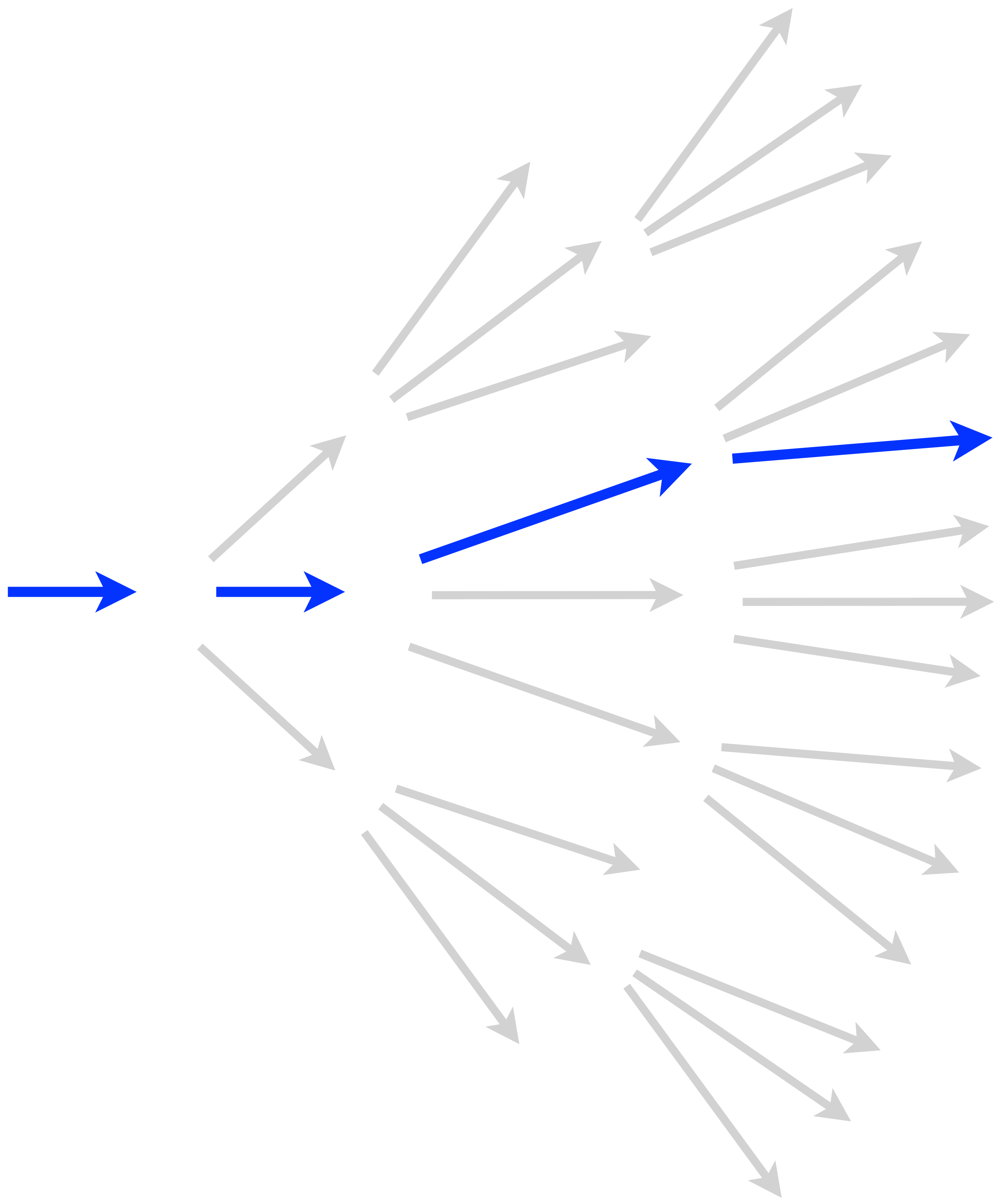
- each state => three new states (shift, l-reduce, r-reduce)
- greedy search: always pick the best next state
  - “best” is defined by a score learned from data

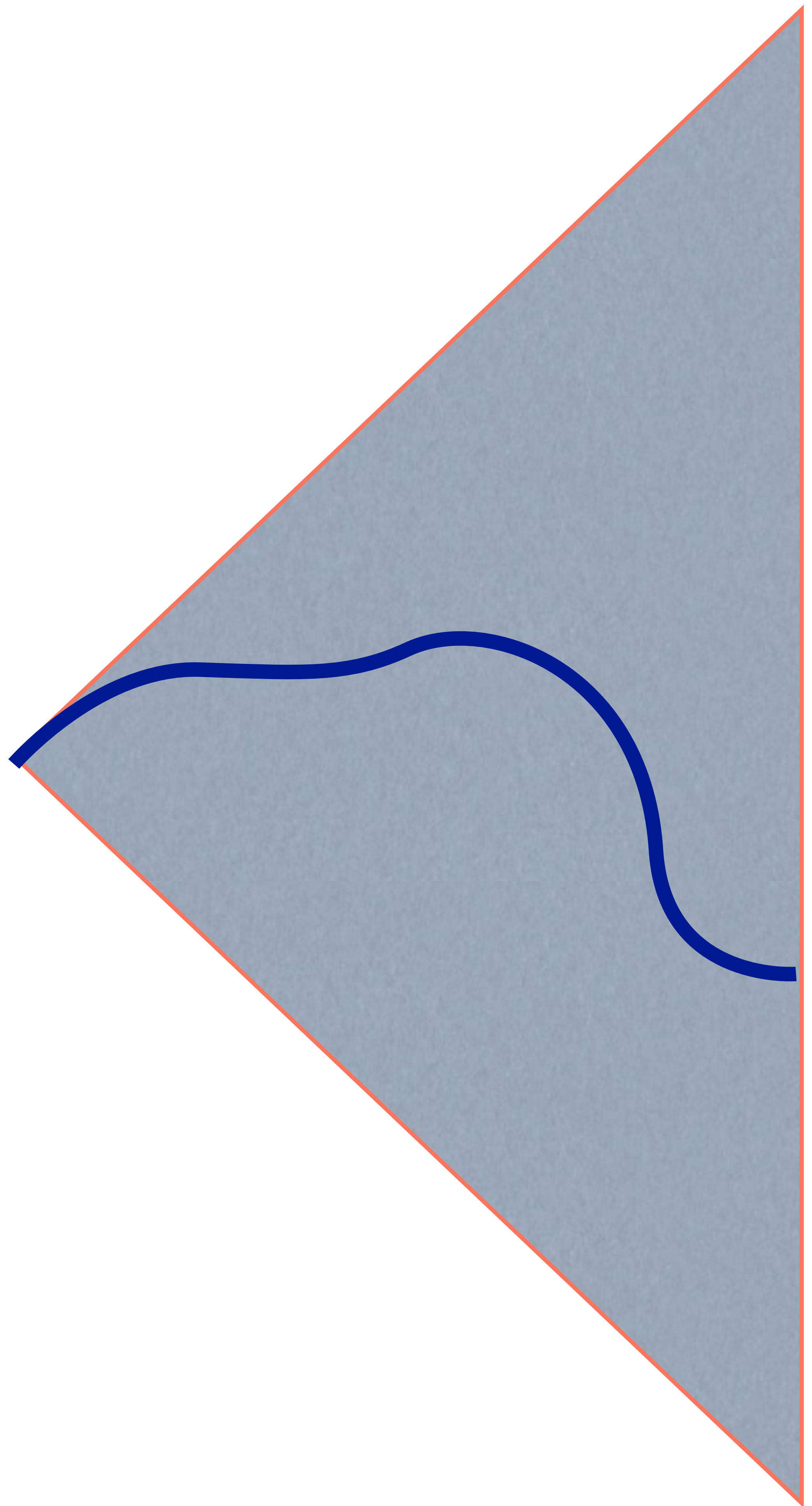




Greedy Search

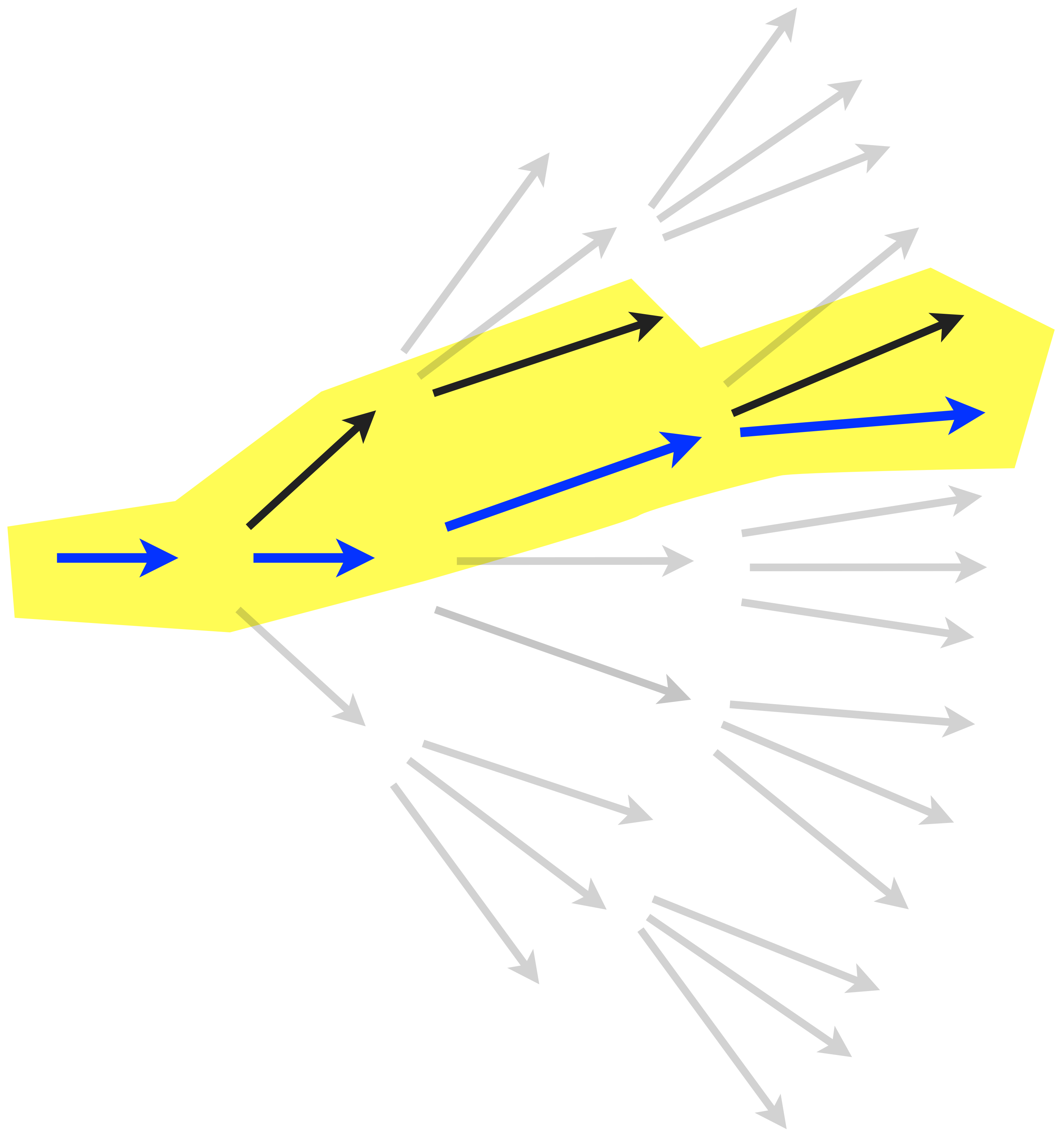
- each state => three new states (shift, l-reduce, r-reduce)
- greedy search: always pick the best next state
  - “best” is defined by a score learned from data

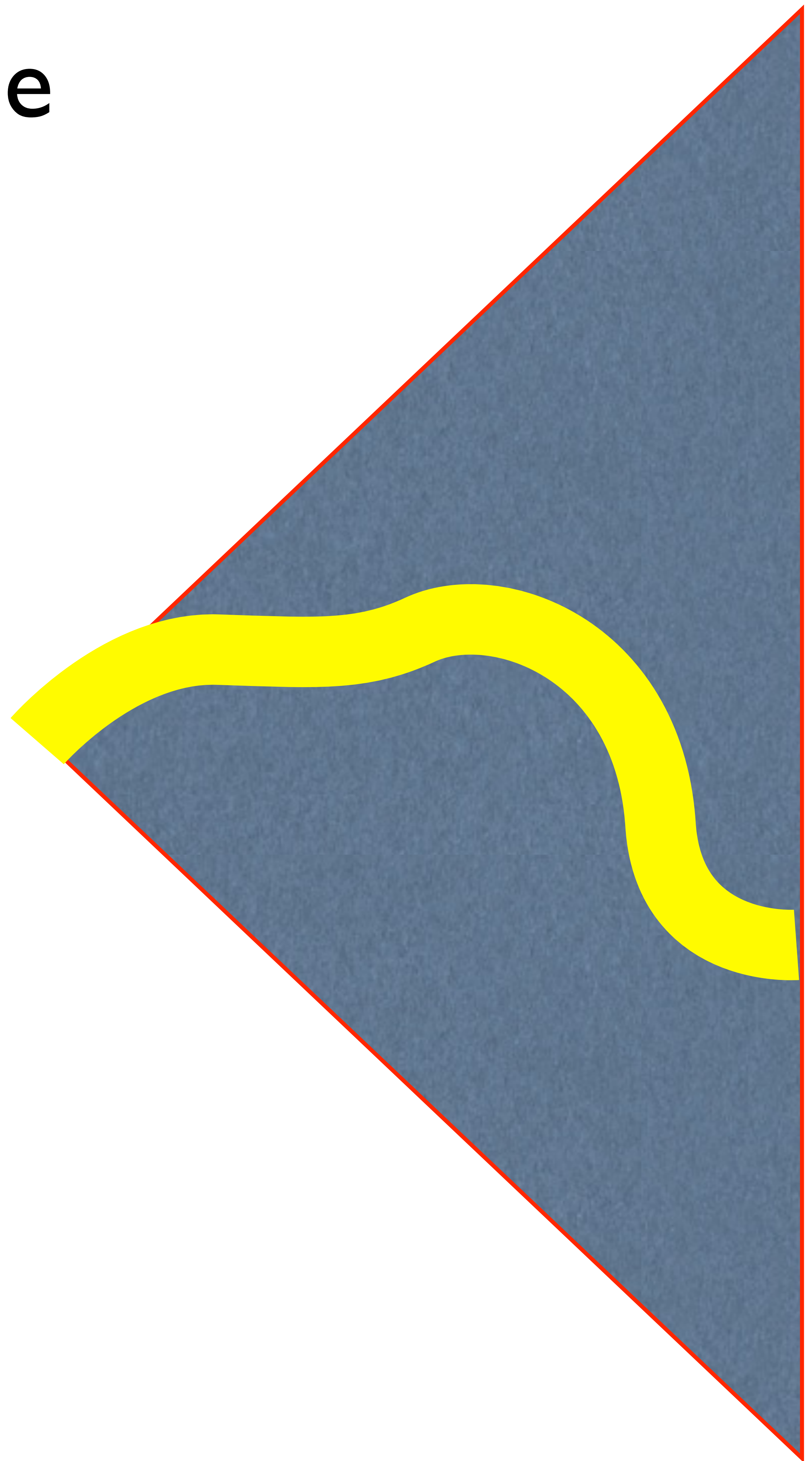




Beam Search

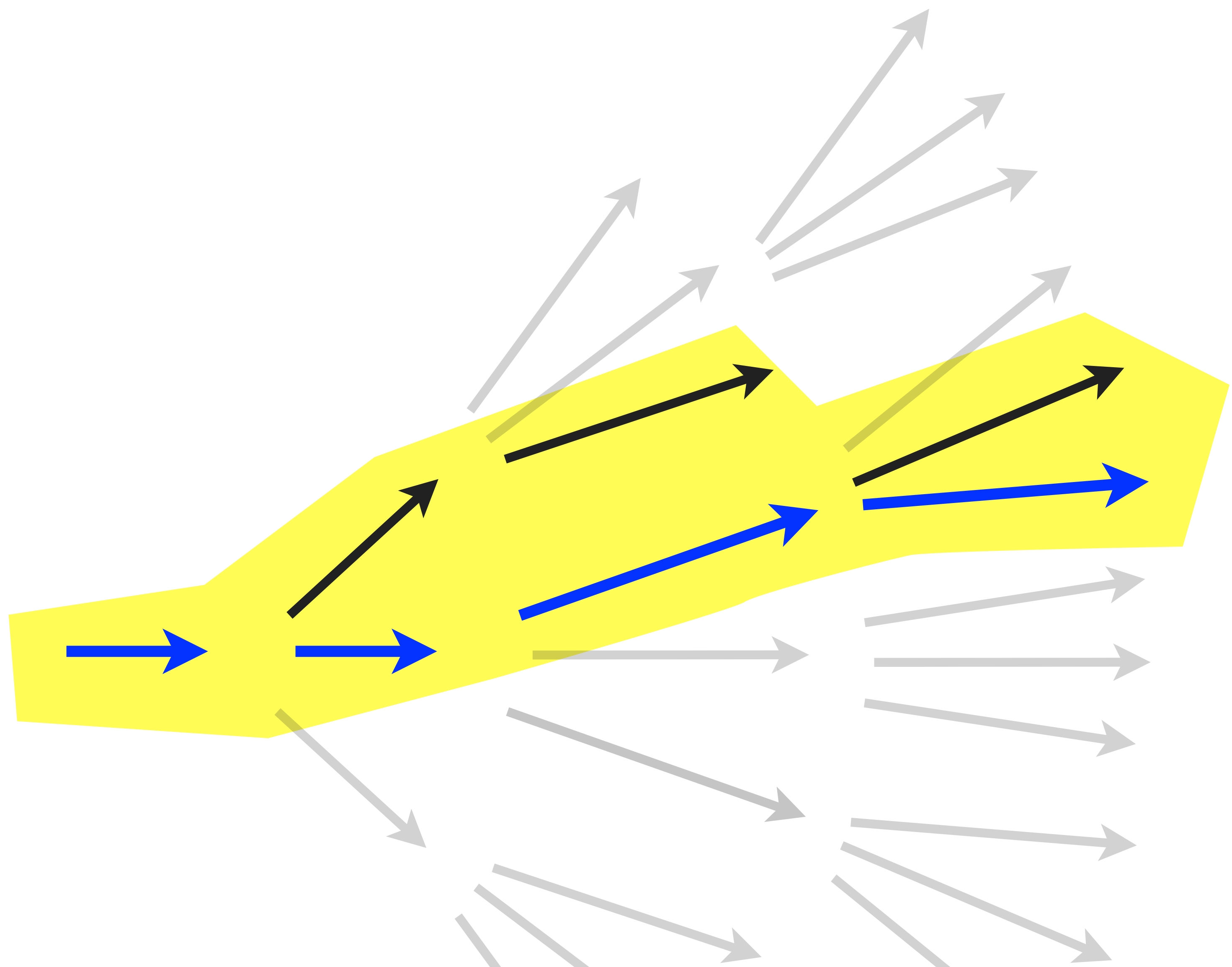
- each state => three new states (shift, l-reduce, r-reduce)
- beam search: always keep top-*b* states
  - still just a tiny fraction of the whole search space

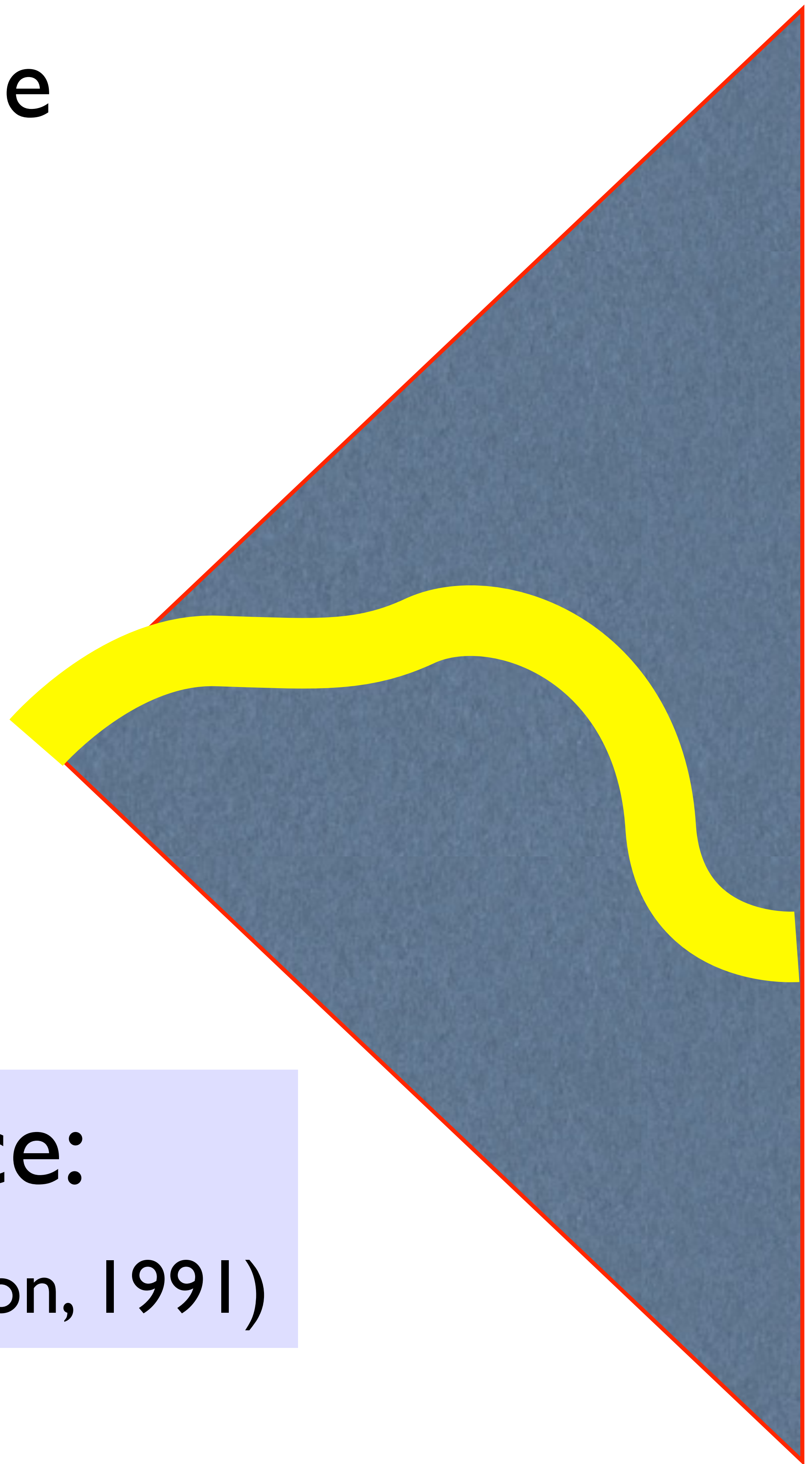




Beam Search

- each state => three new states (shift, l-reduce, r-reduce)
- beam search: always keep top-*b* states
  - still just a tiny fraction of the whole search space

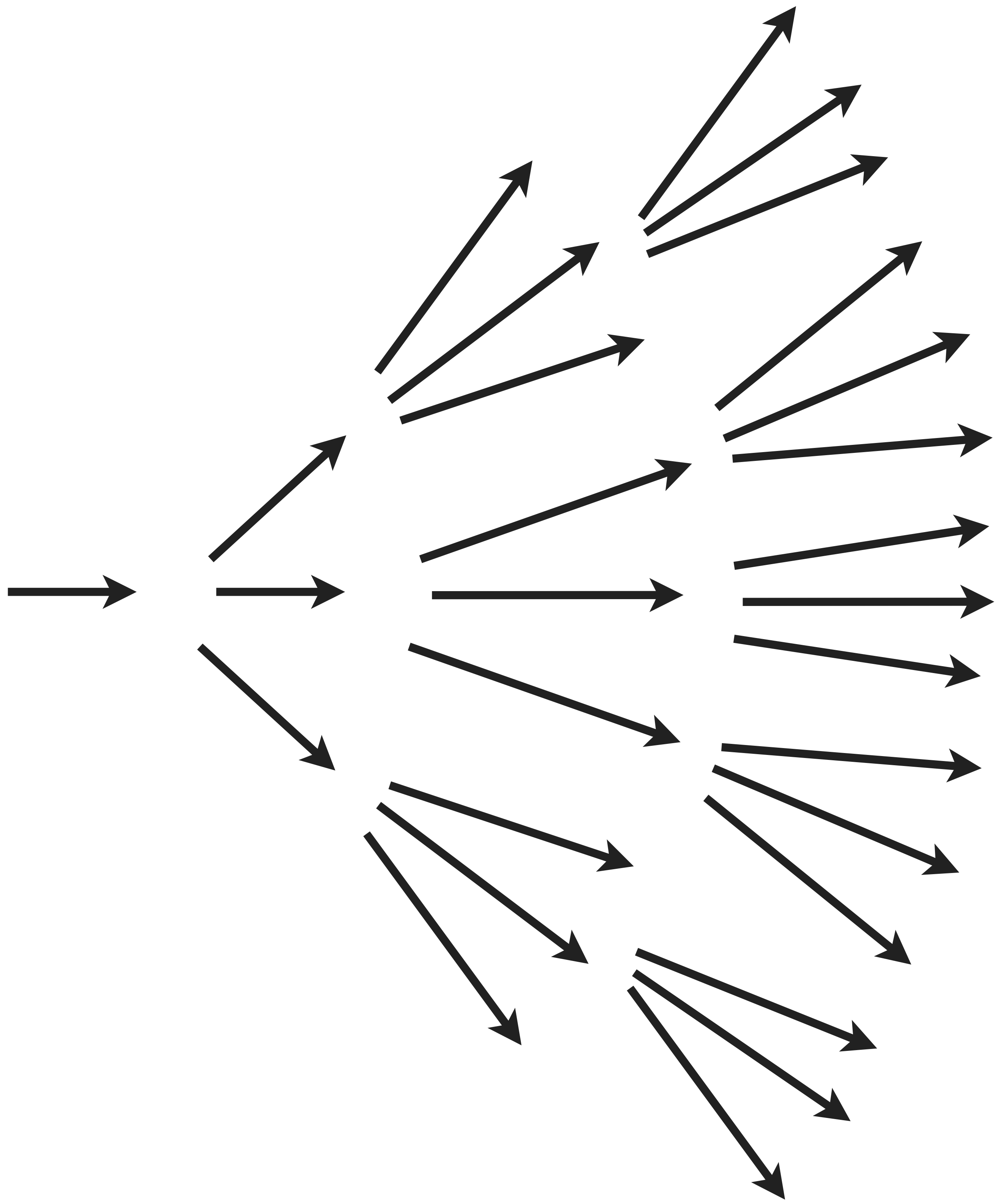


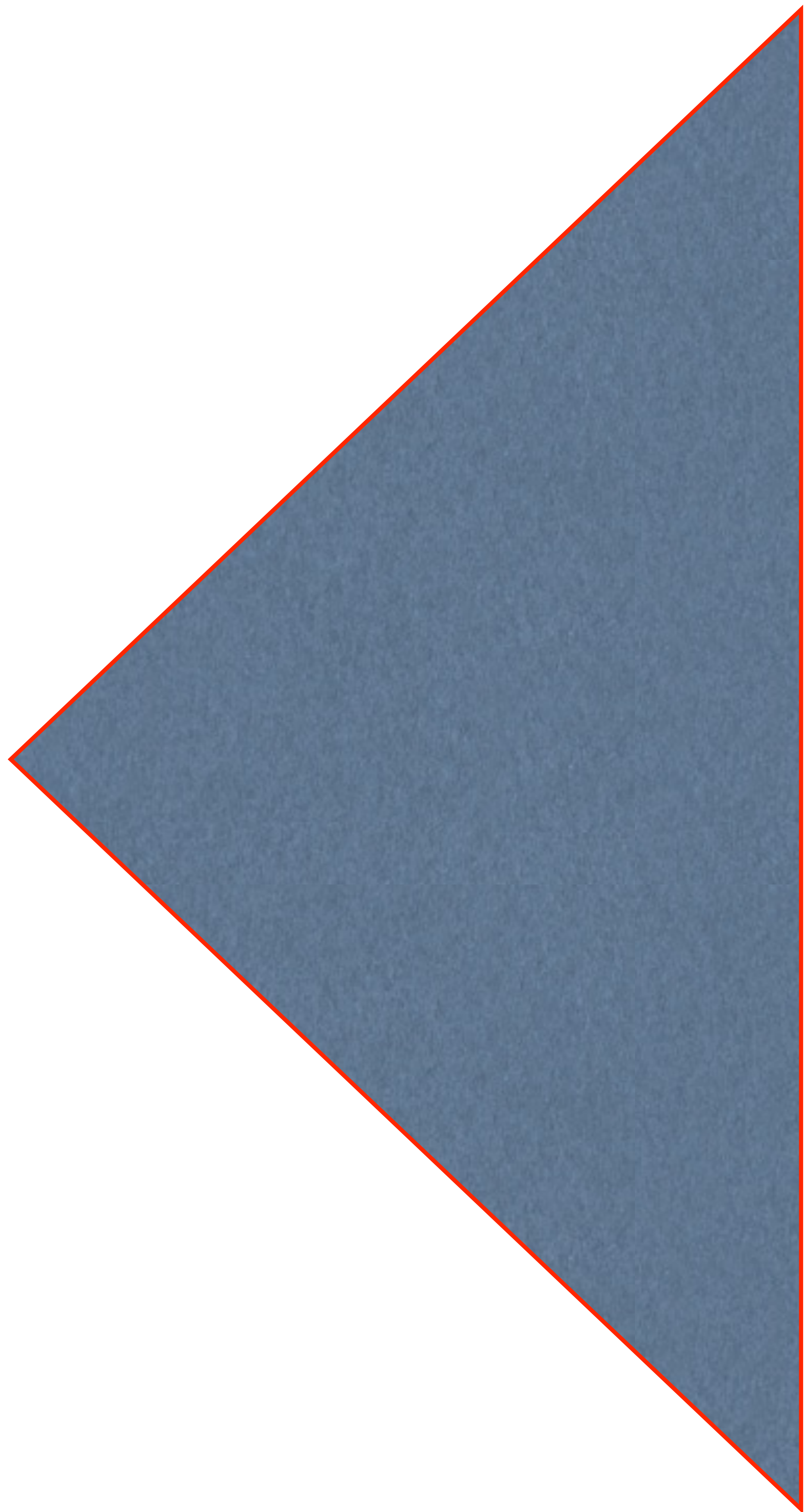


psycholinguistic evidence:
parallelism (Fodor et al, 1974; Gibson, 1991)

Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space

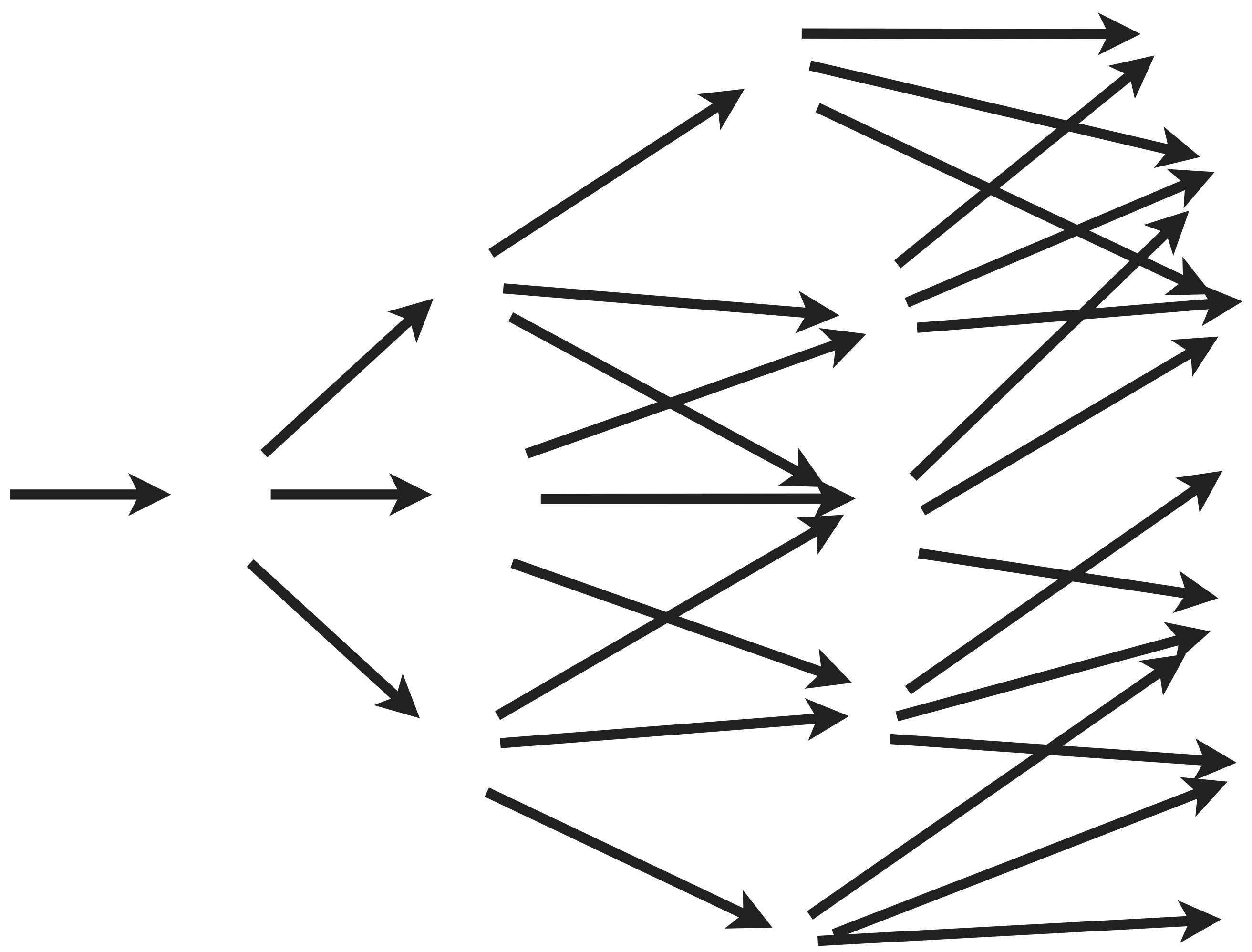


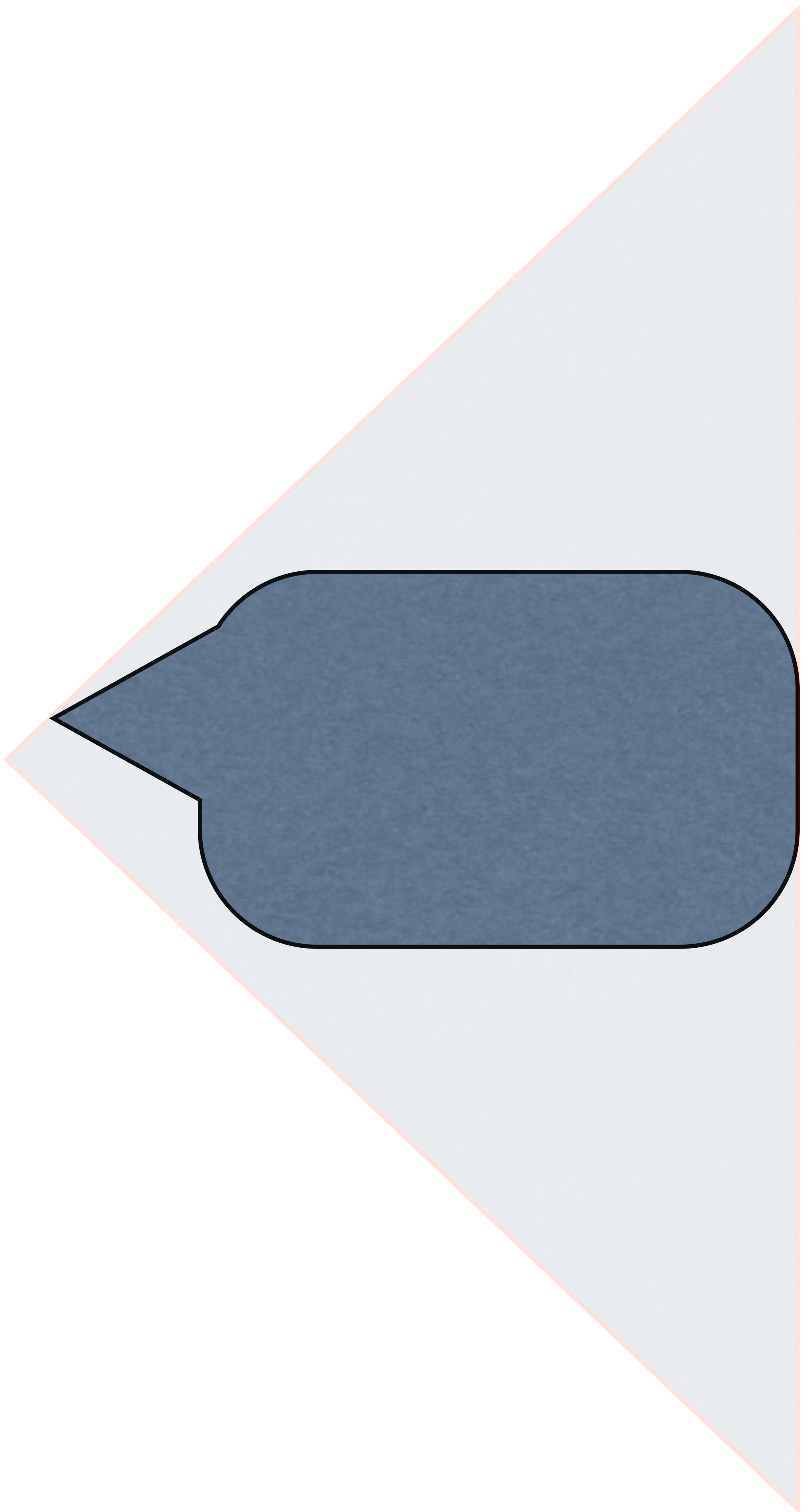


(Huang and Sagae, 2010)

Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space

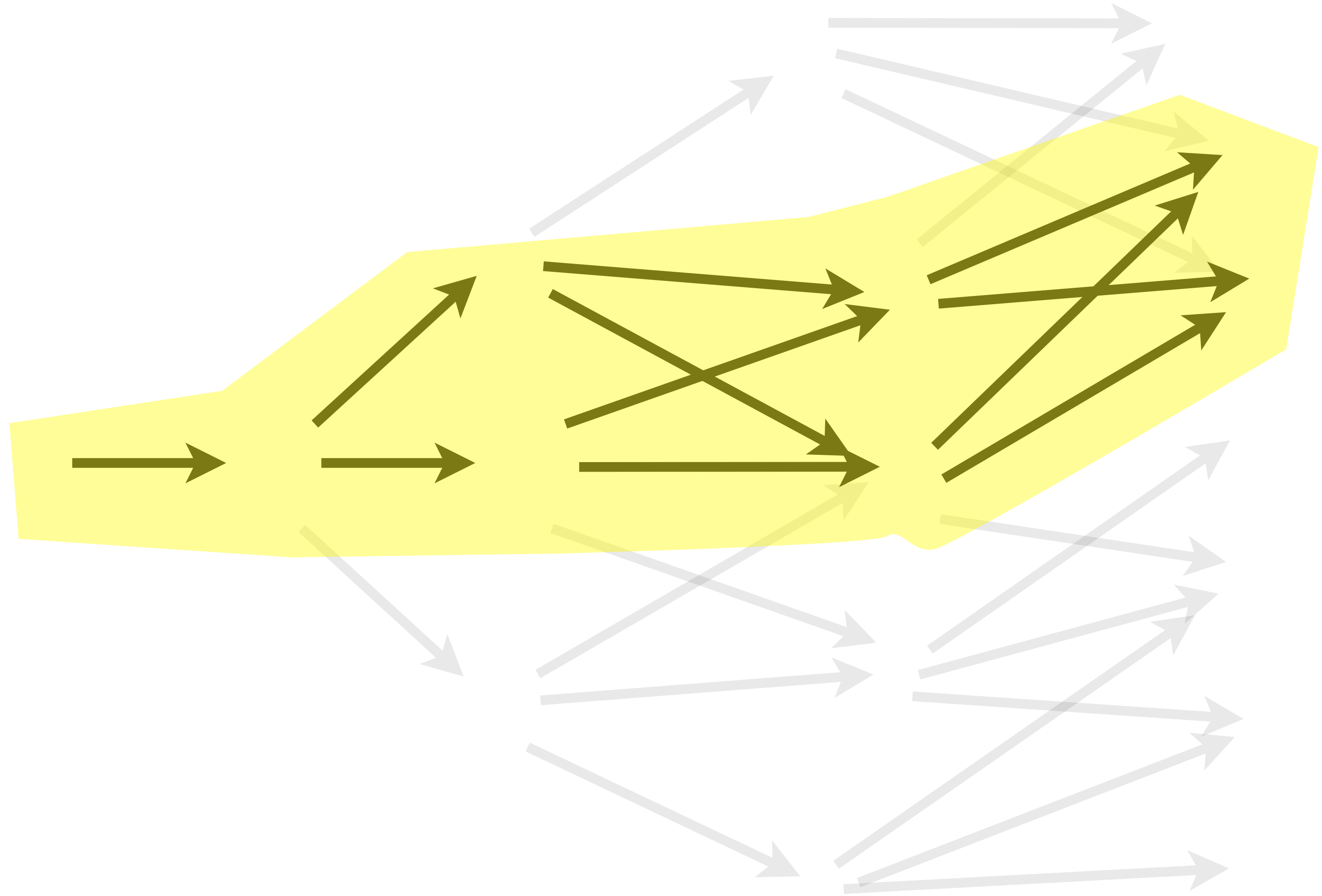


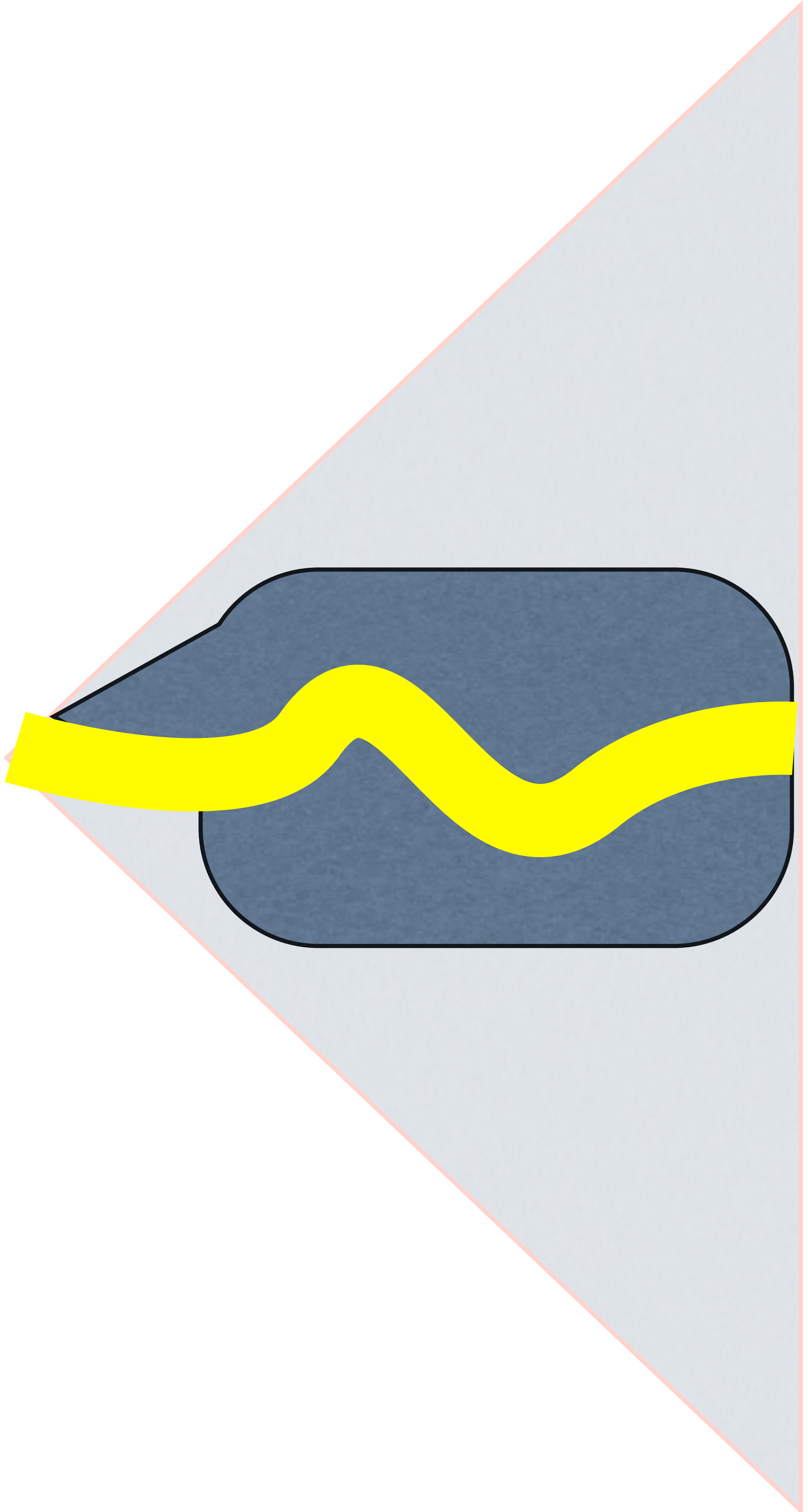


(Huang and Sagae, 2010)

Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space



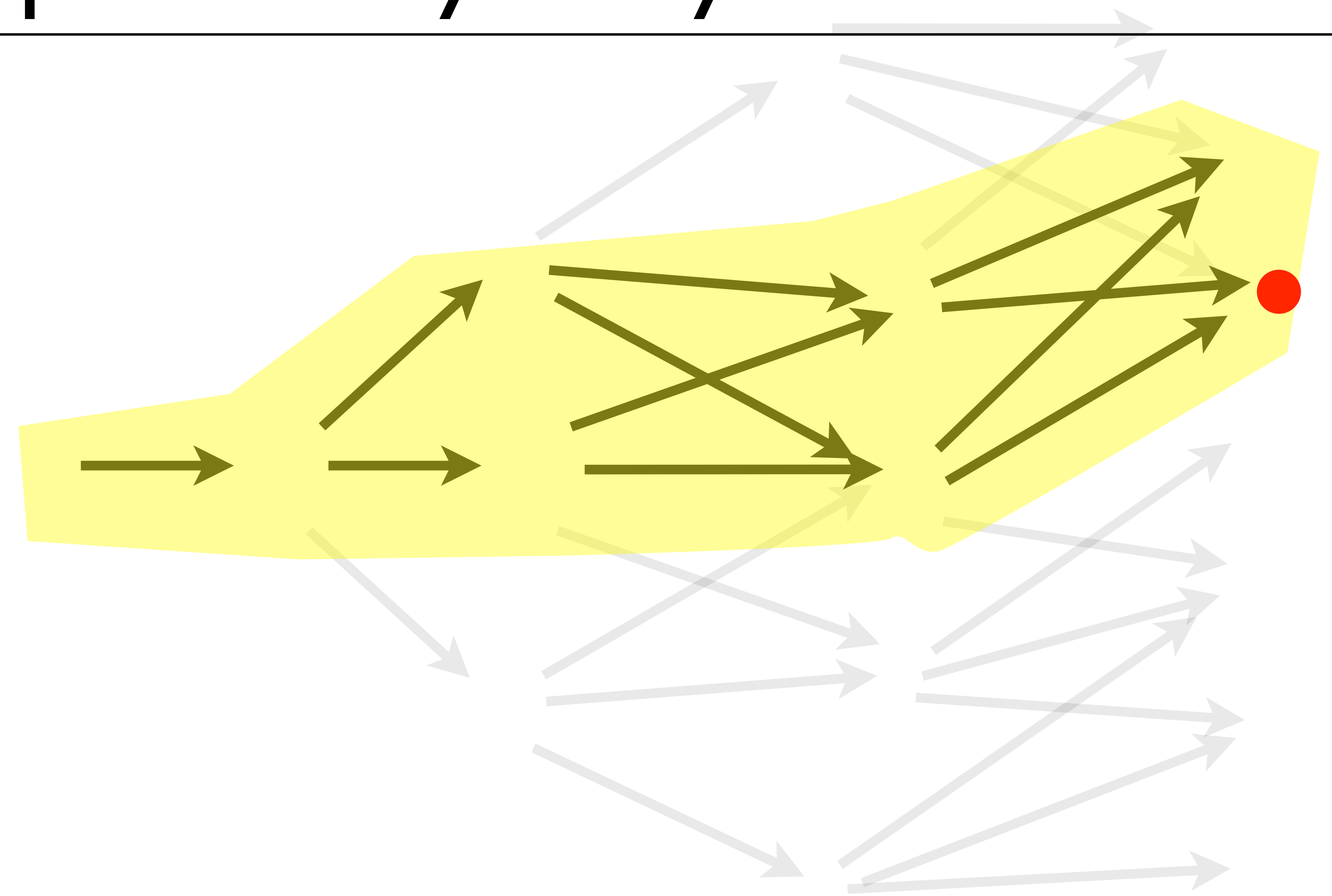


(Huang and Sagae, 2010)

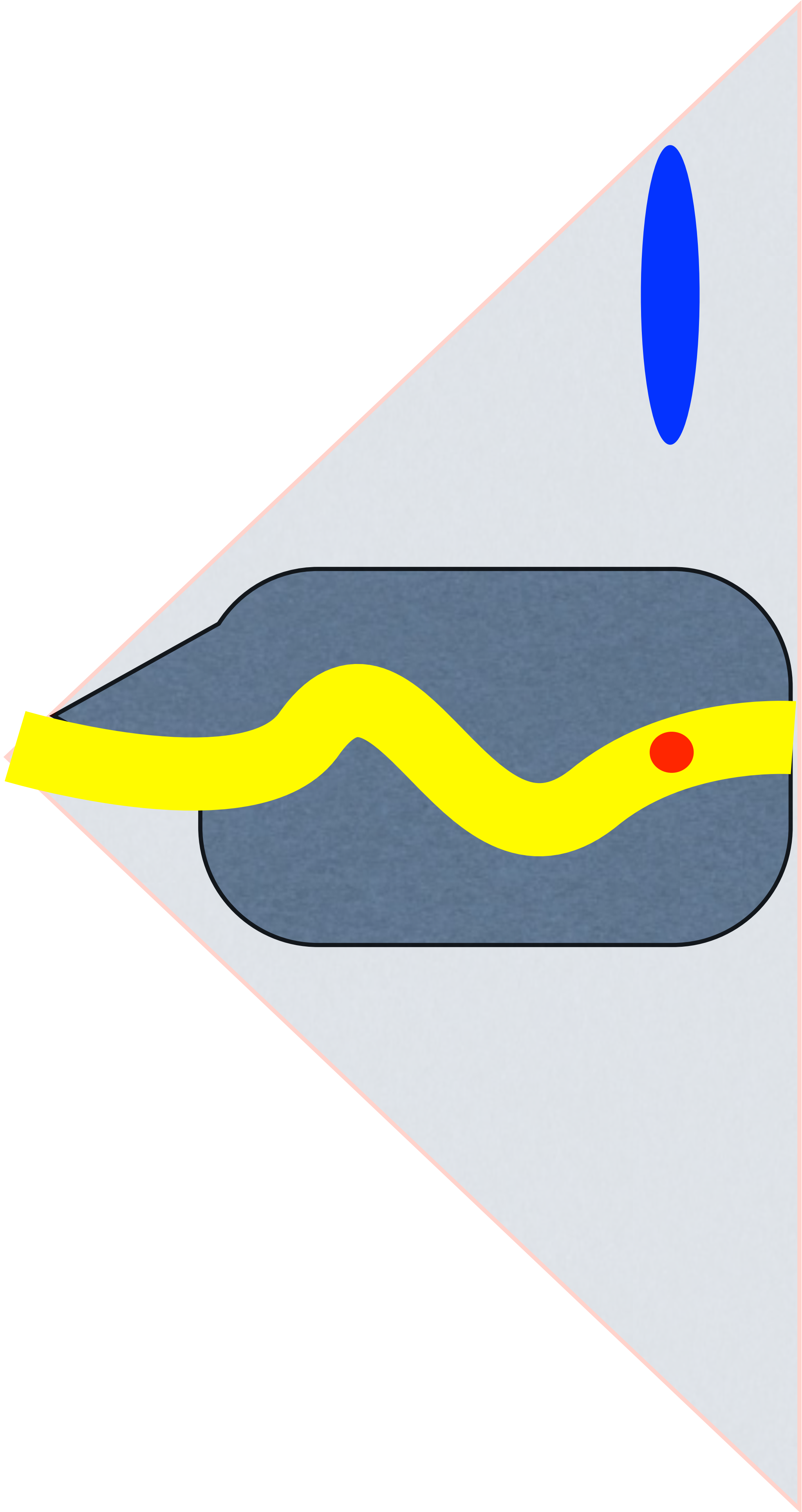
Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space

each DP state corresponds to
exponentially many non-DP states



graph-structured stack
(Tomita, 1986)

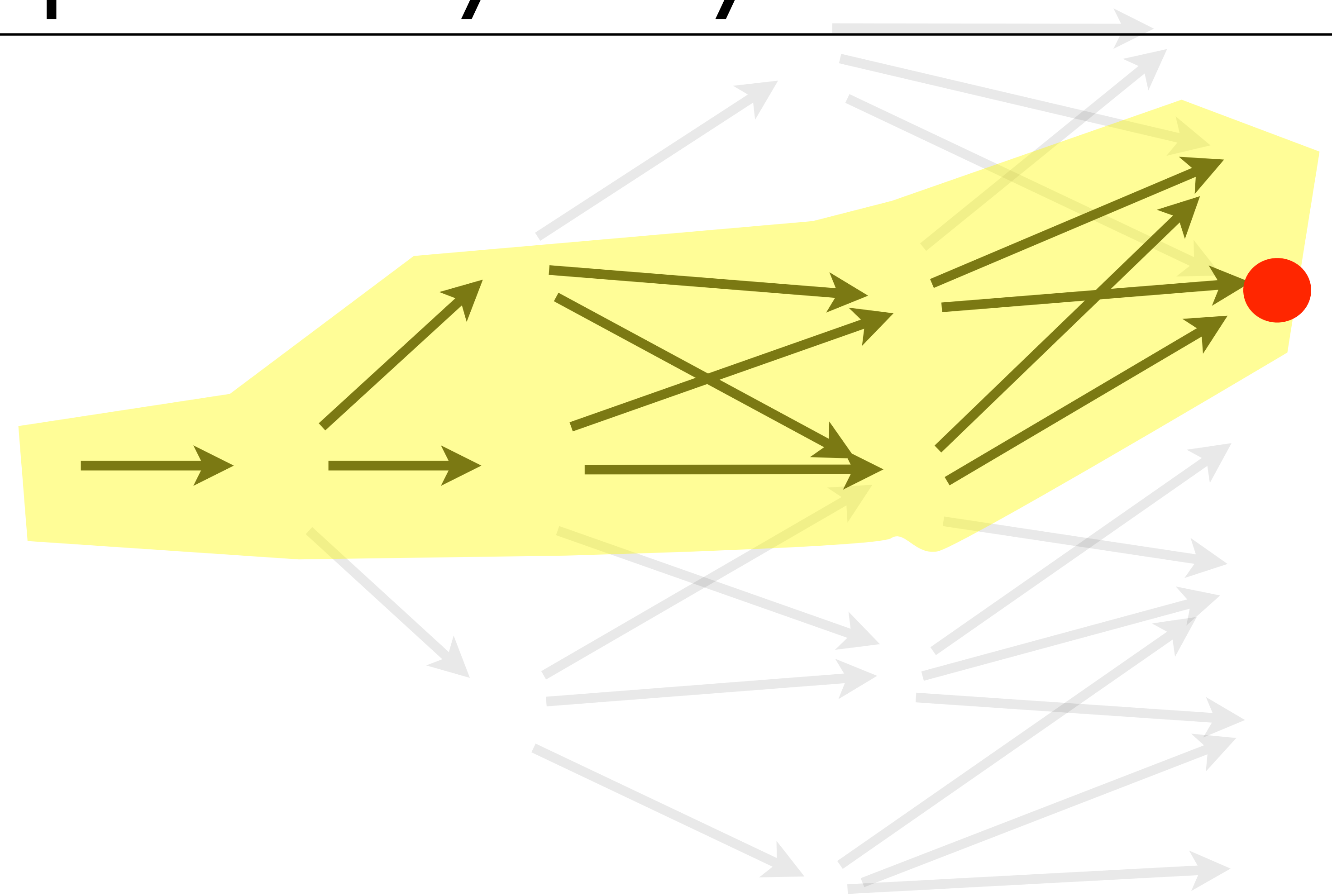


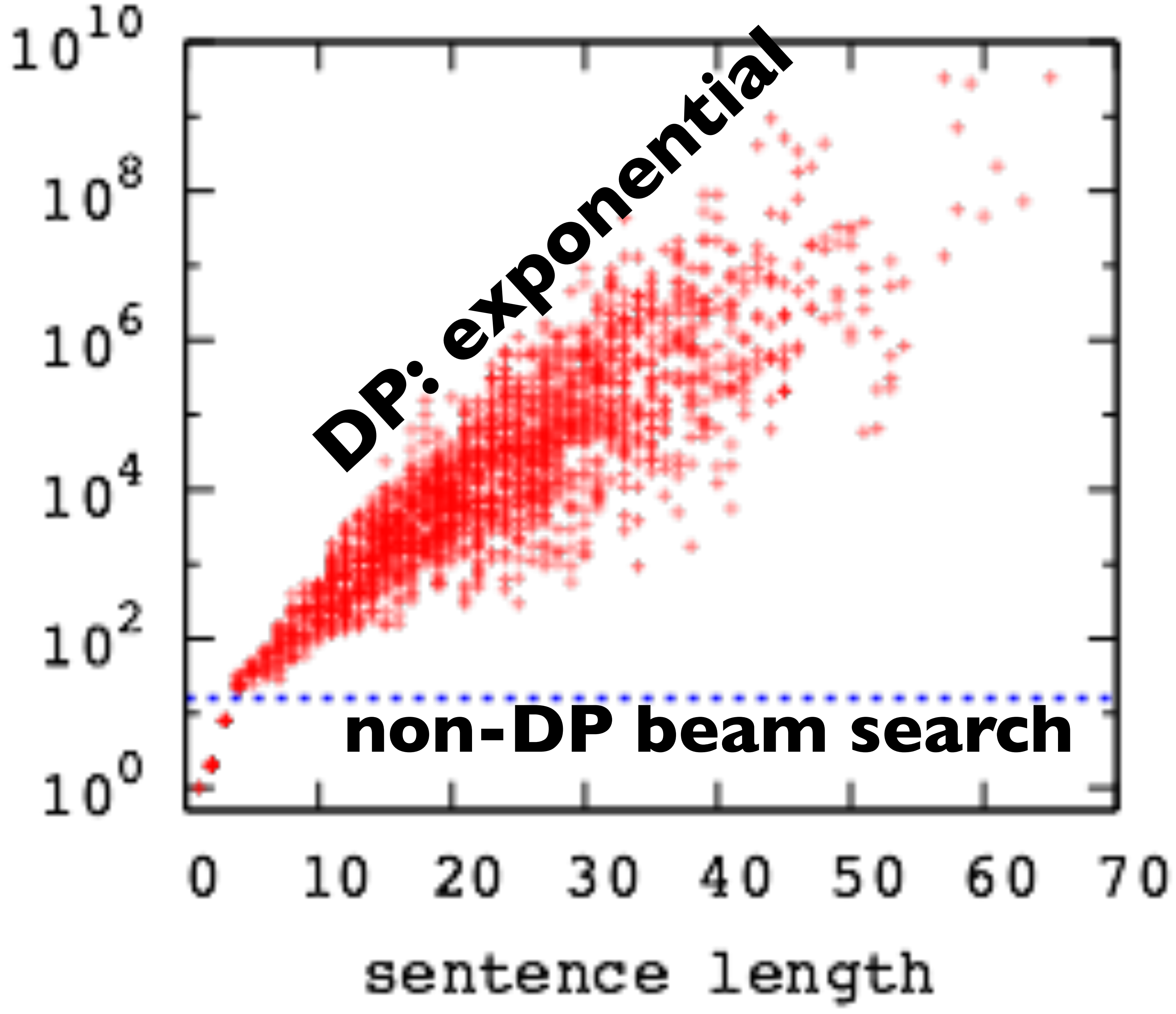
(Huang and Sagae, 2010)

Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space

each DP state corresponds to exponentially many non-DP states



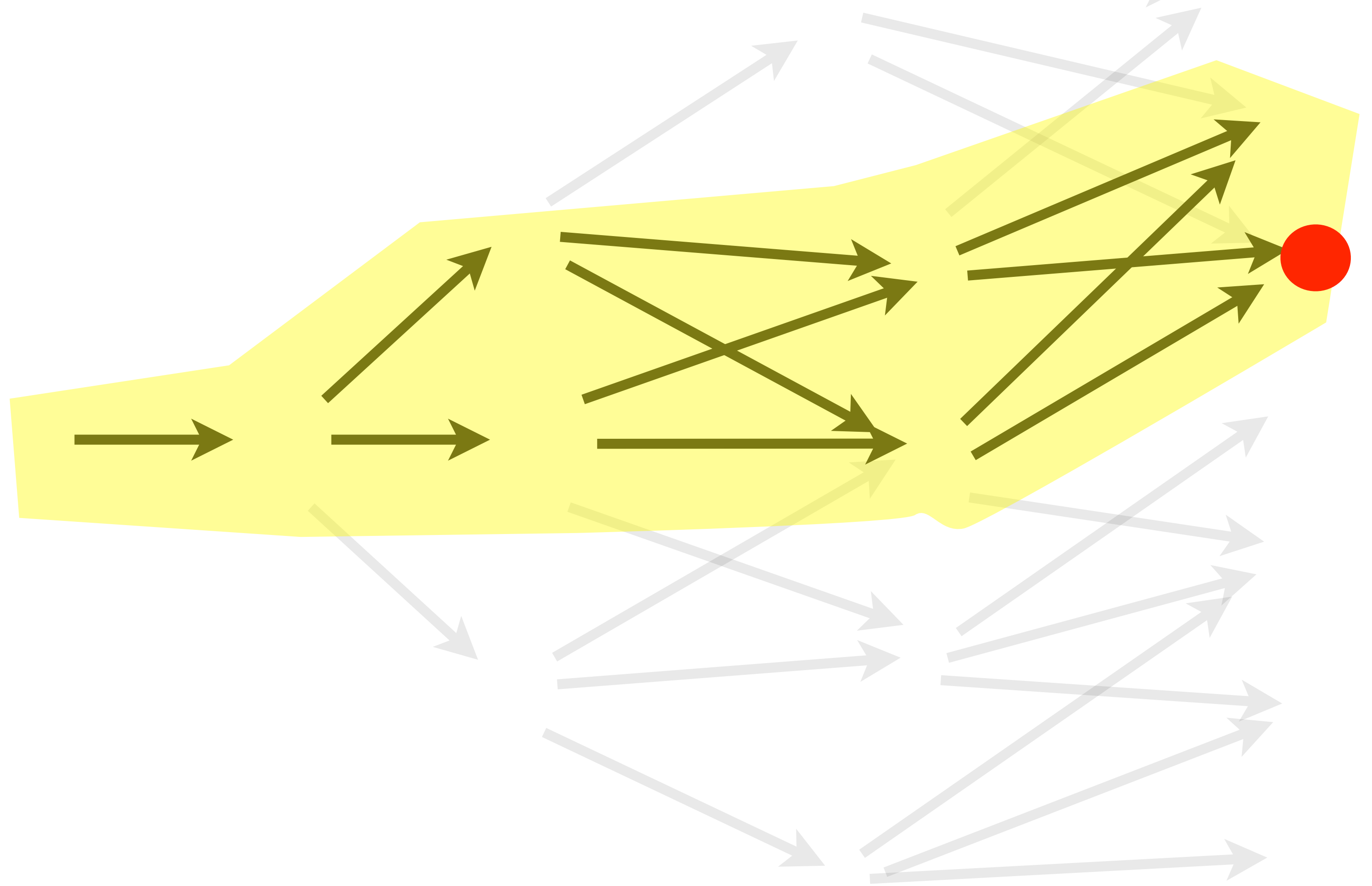


(Huang and Sagae, 2010)

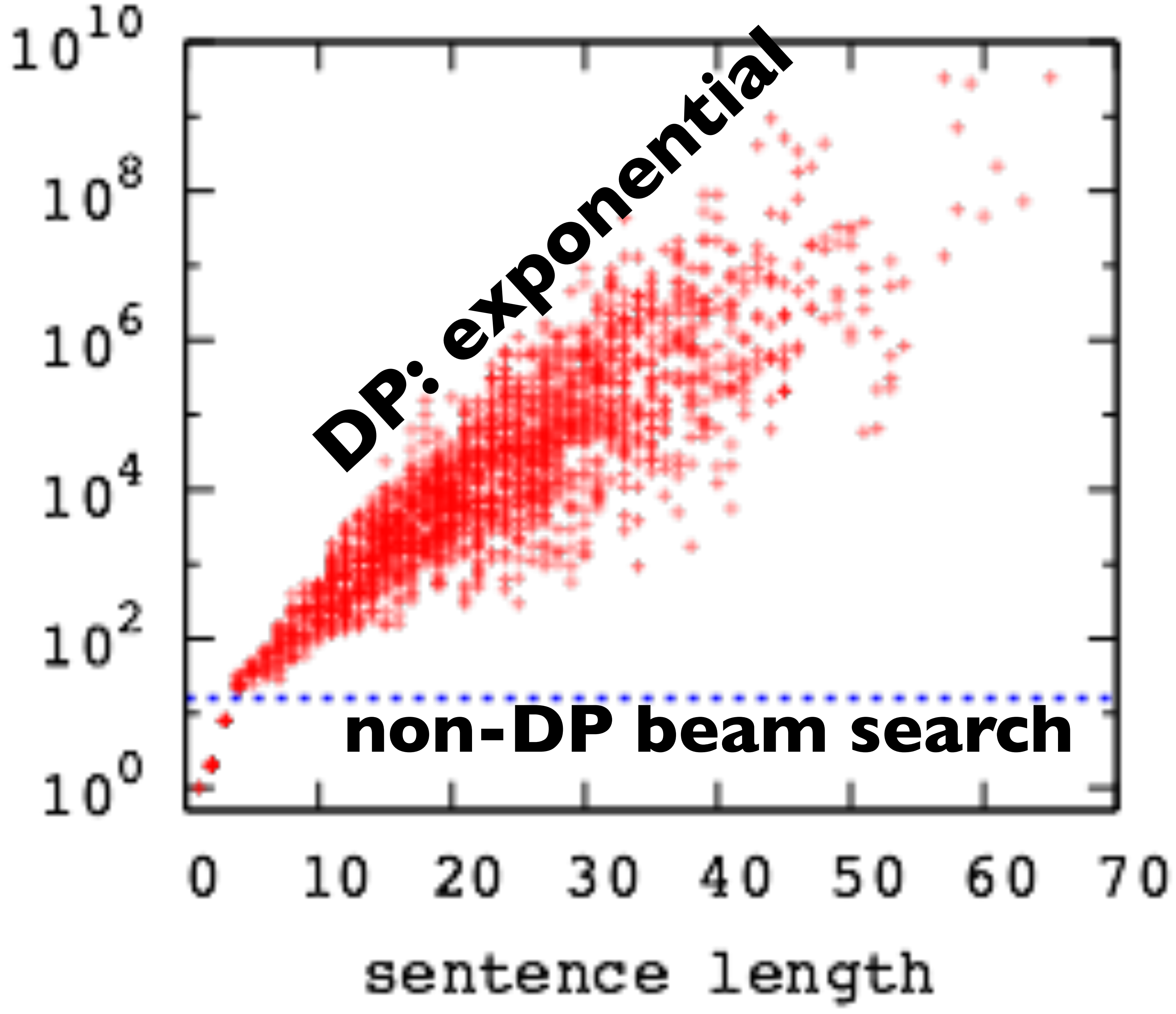
Dynamic Programming

- each state => three new states (shift, l-reduce, r-reduce)
- key idea of DP: share common subproblems
  - merge equivalent states => polynomial space

each DP state corresponds to exponentially many non-DP states



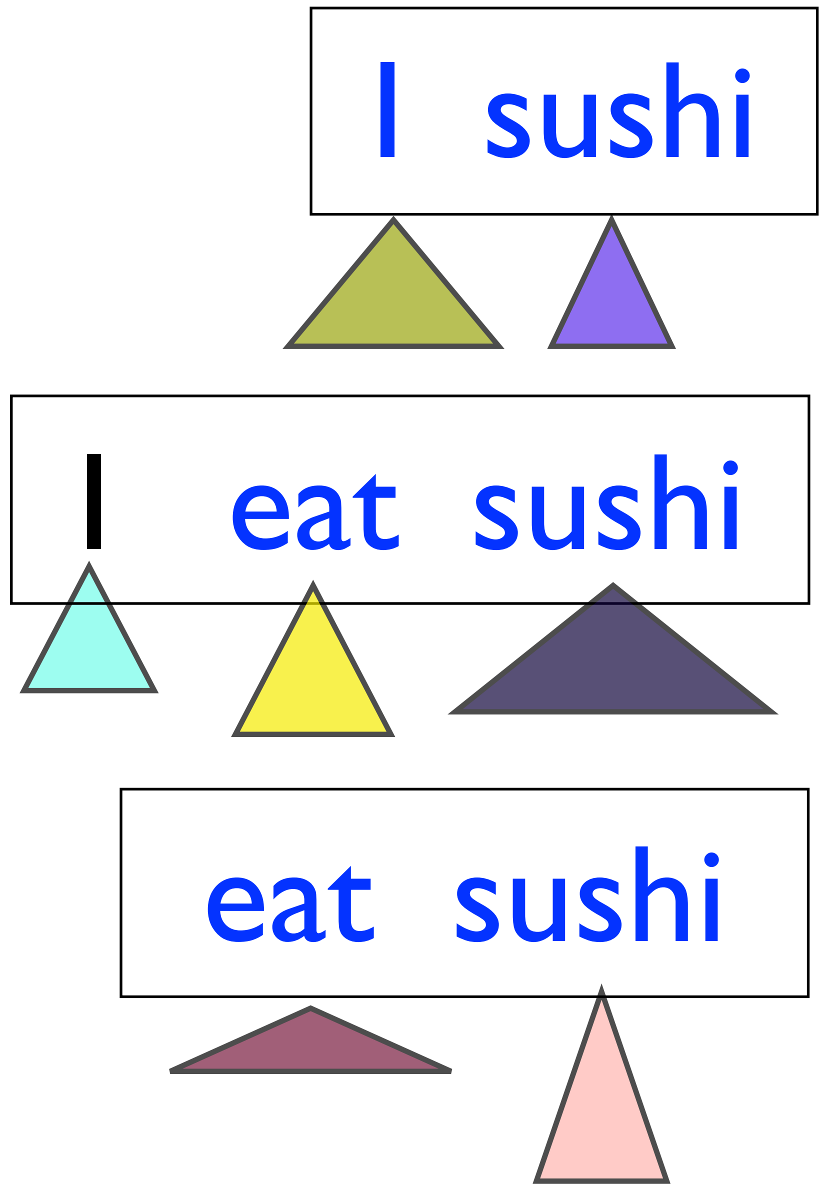
graph-structured stack
(Tomita, 1986)



(Huang and Sagae, 2010)

Merging (Ambiguity Packing)

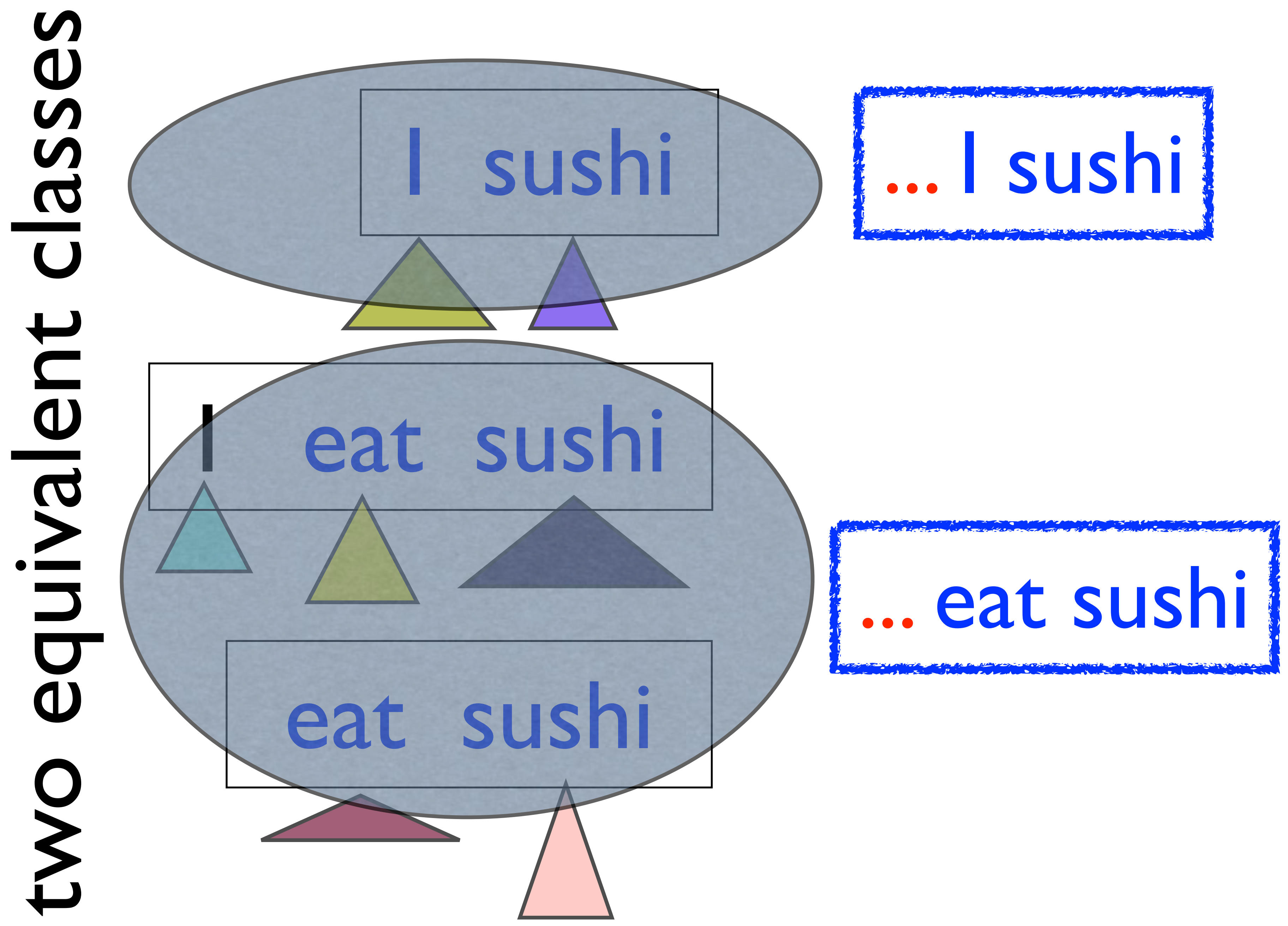
- two states are equivalent if they agree on features
  - because same features guarantee same cost
  - example: if we only care about the last 2 words on stack



(Huang and Sagae, 2010)

Merging (Ambiguity Packing)

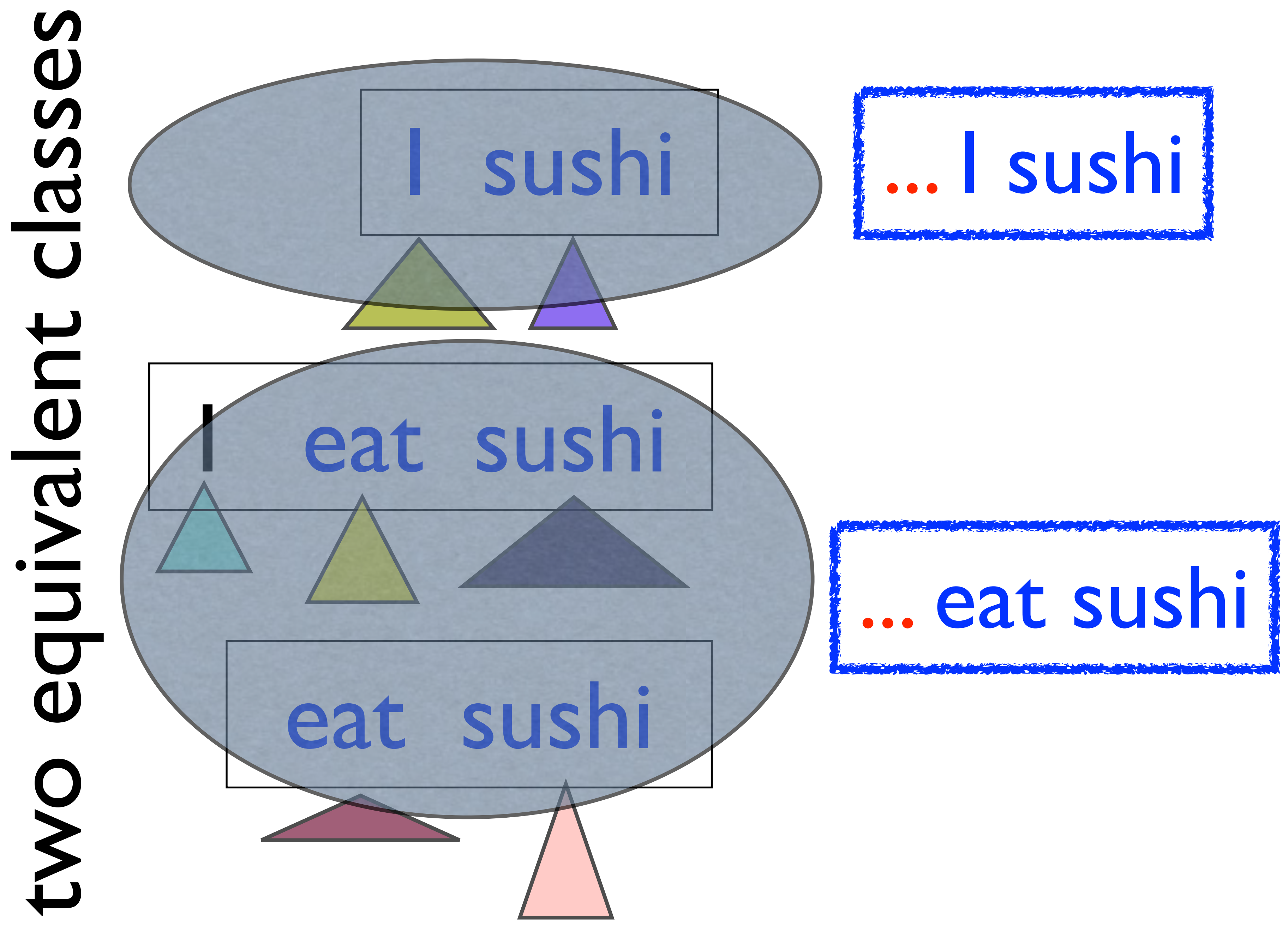
- two states are equivalent if they agree on features
  - because same features guarantee same cost
  - example: if we only care about the last 2 words on stack



(Huang and Sagae, 2010)

Merging (Ambiguity Packing)

- two states are equivalent if they agree on features
  - because same features guarantee same cost
  - example: if we only care about the last 2 words on stack



psycholinguistic evidence
(eye-tracking experiments):

delayed disambiguation

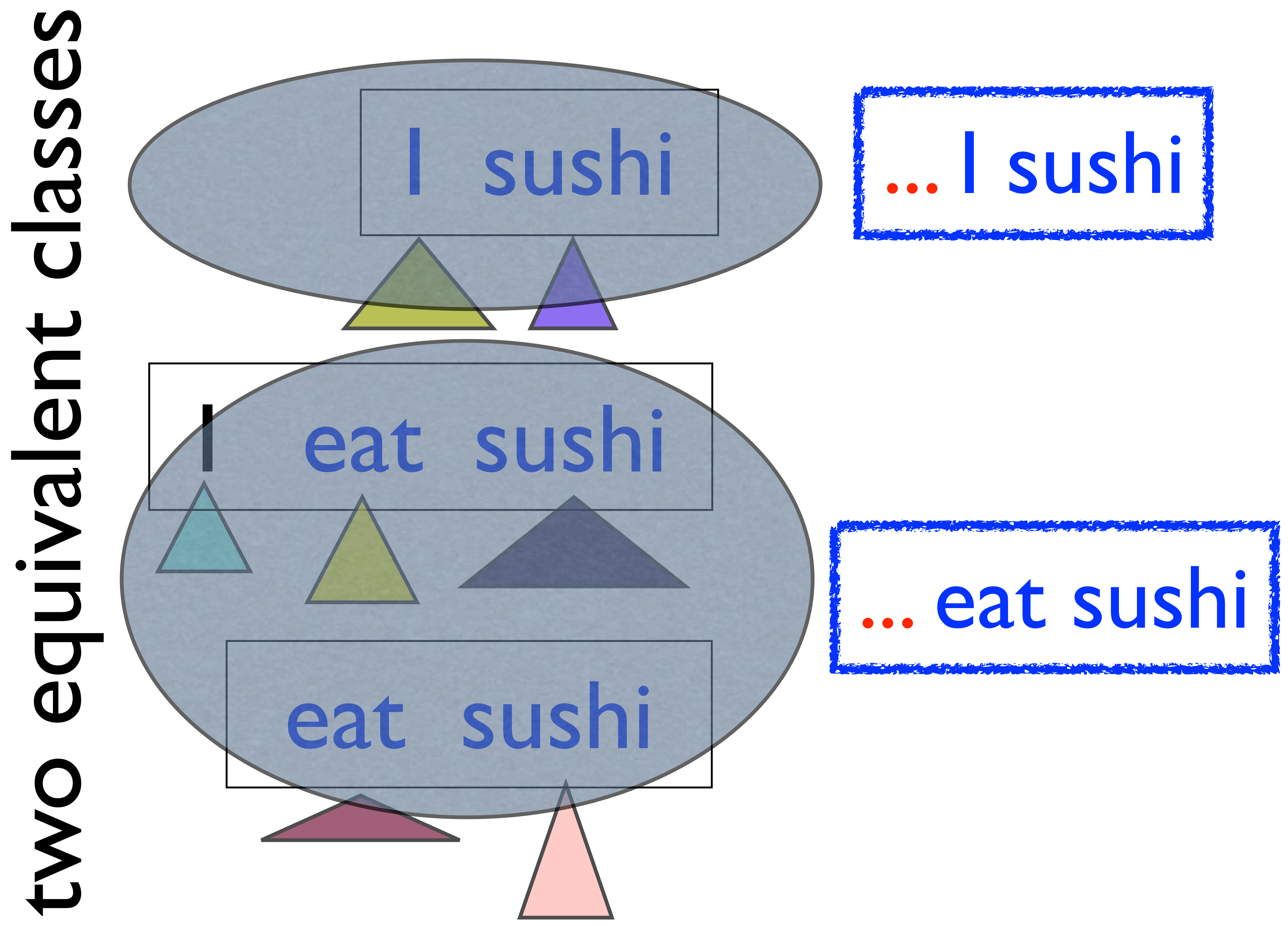
John and Mary had 2 papers
John and Mary had 2 papers

Frazier and Rayner (1990), Frazier (1999)

(Huang and Sagae, 2010)

Merging (Ambiguity Packing)

- two states are equivalent if they agree on features
  - because same features guarantee same cost
  - example: if we only care about the last 2 words on stack



psycholinguistic evidence
(eye-tracking experiments):

delayed disambiguation

John and Mary had 2 papers each
John and Mary had 2 papers together

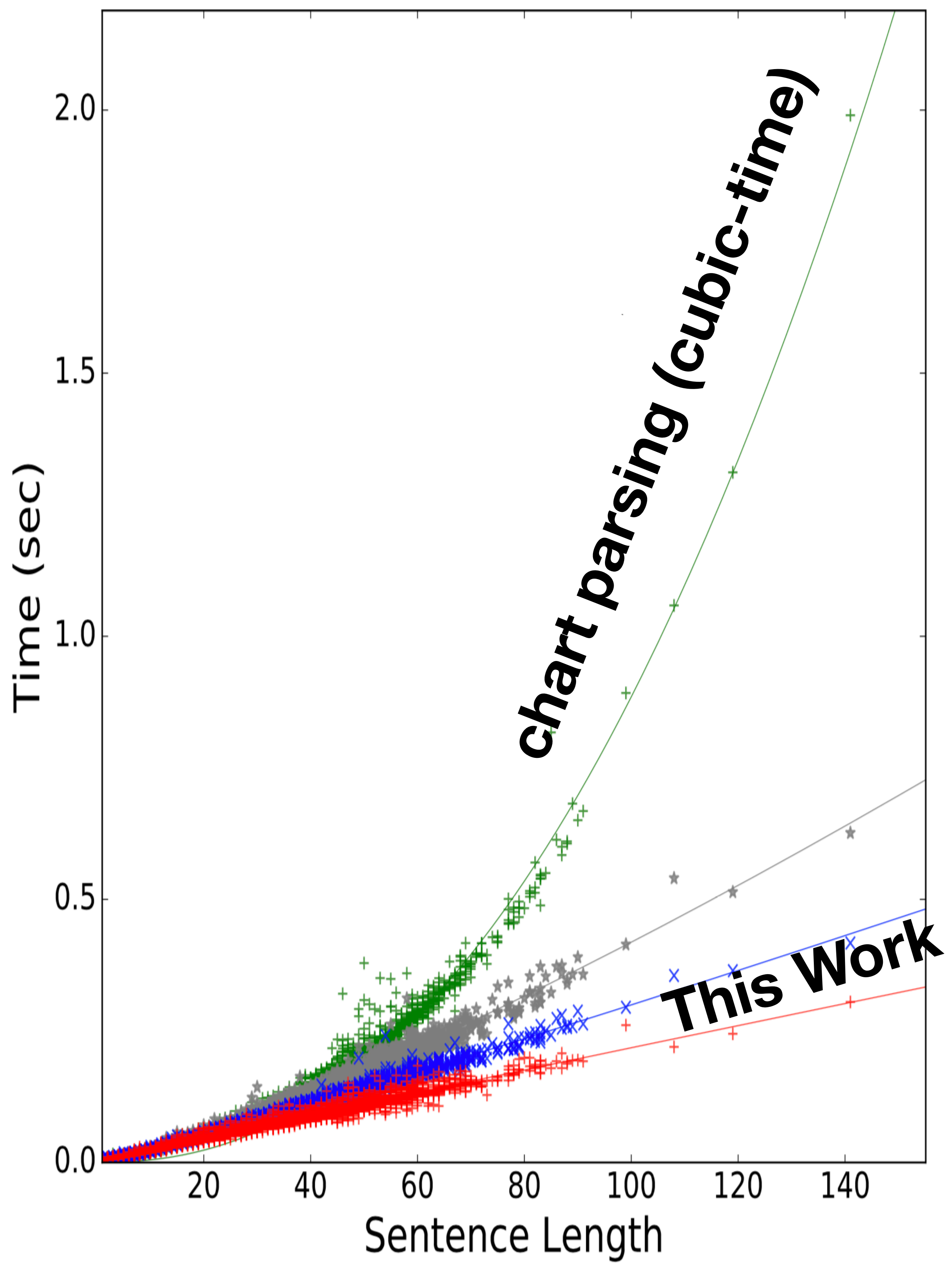
Frazier and Rayner (1990), Frazier (1999)

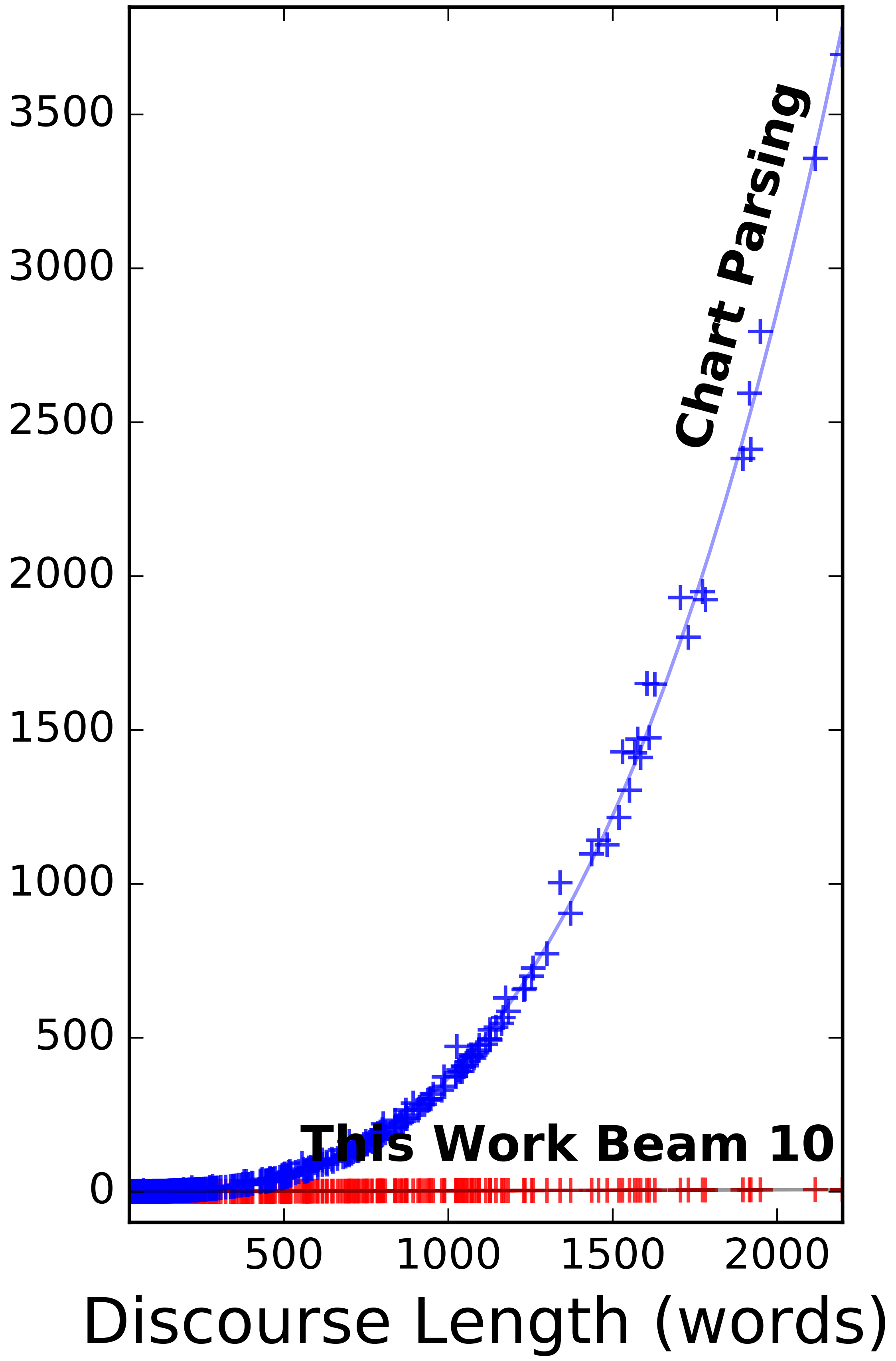
(Huang and Sagae, 2010)

Results: Fast and Accurate

Constituency parsing, PTB only, Single Model, End-to-End

Parsre	Note	F1 Score
Durett + Klein 2015	cubic-time parser	91.1
Cross + Huang 2016	original span parser (greedy)	91.3
Liu + Zhang 2016	greedy / beam	91.7
Dyer et al. 2016	greedy / beam	91.7
Stern 2017a	cubic-time span-based parser	91.79
Our Work	linear-time dynamic programming, span-based	**91.97**





(Hong and Huang, 2018)

Part III: Linear-Time RNA Structure Prediction

(Huang et al, 2019; under review)

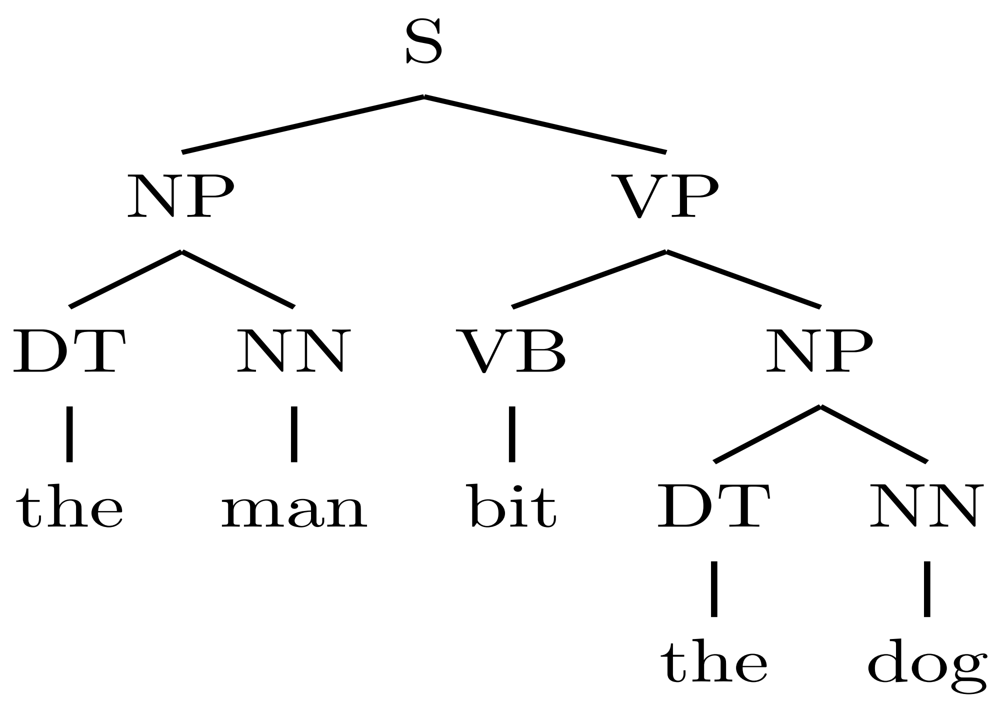
Part III: Linear-Time RNA Structure Prediction

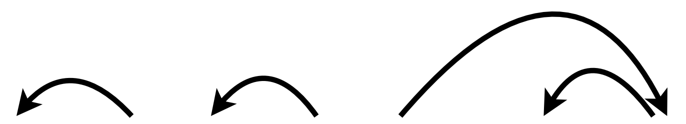
(Huang et al, 2019; under review)

Computational Linguistics => Computational Biology

linguistics

1955 Chomsky:
context-free grammars





the man bit the dog

computer science

1958 Backus & Naur:
CFGs in programming lang.

1960s CKY Parsing: **$O(n^3)$**

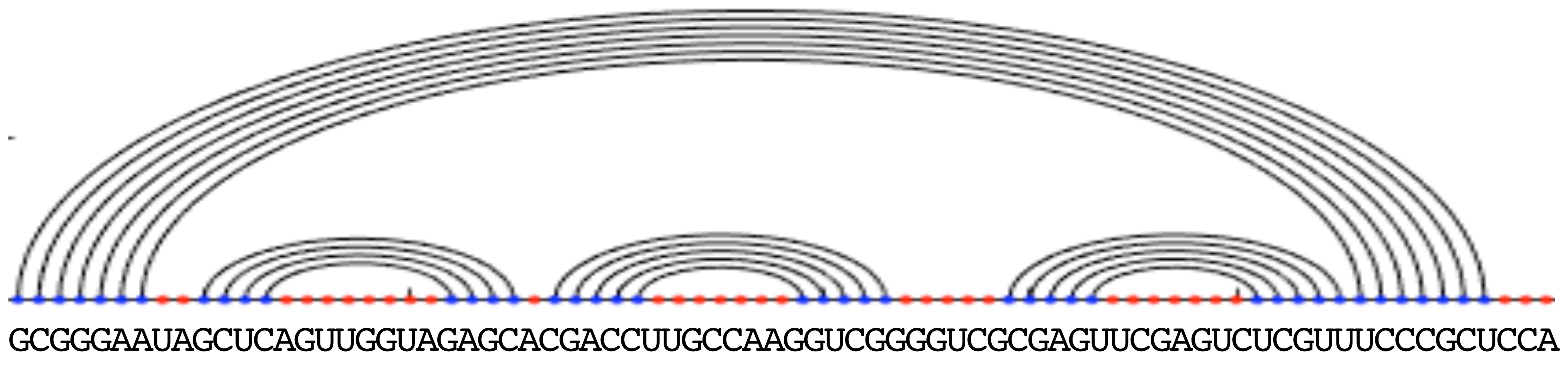
1965 Knuth: LR Parsing: **$O(n)$**

1986 Tomita: Generalized LR Parsing

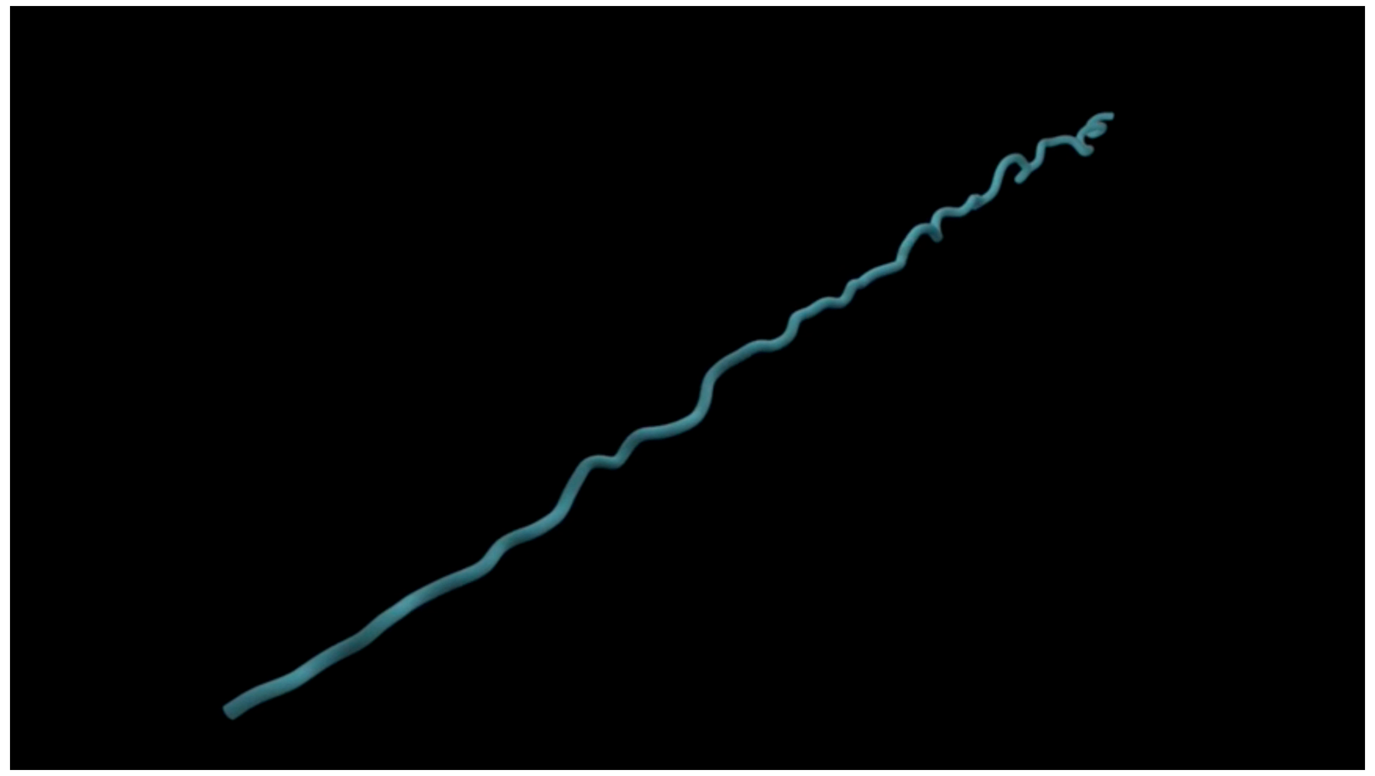
2010: linear-time DP parsing
(Huang & Sagae)

biology

1953 Watson & Crick: 
DNA double-helix



1980s: $O(n^3)$ CKY for RNA structures

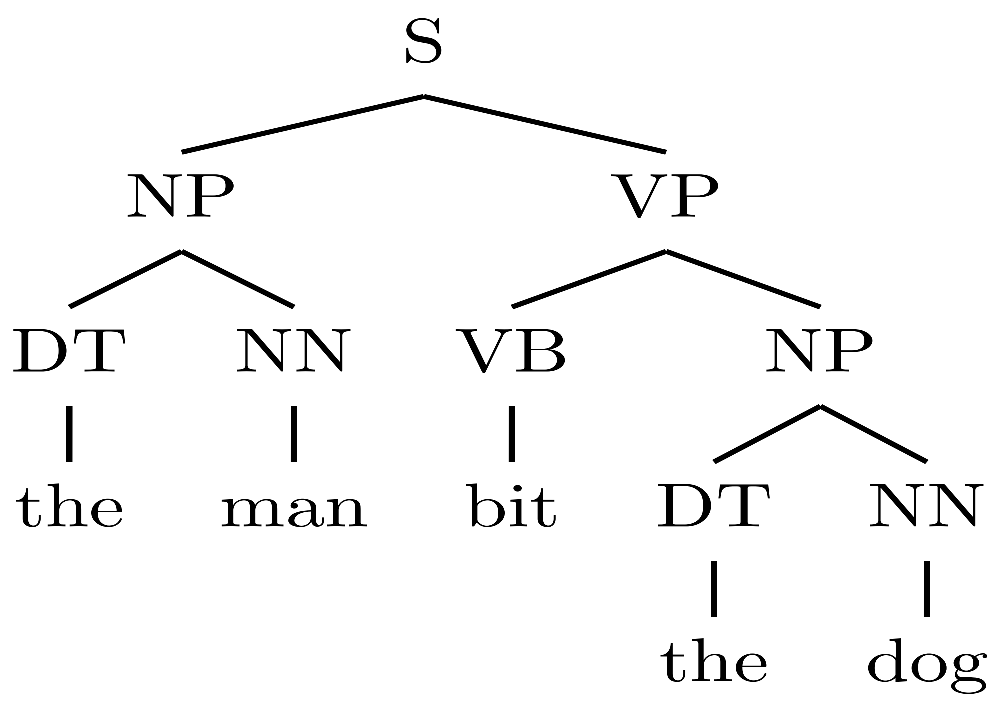


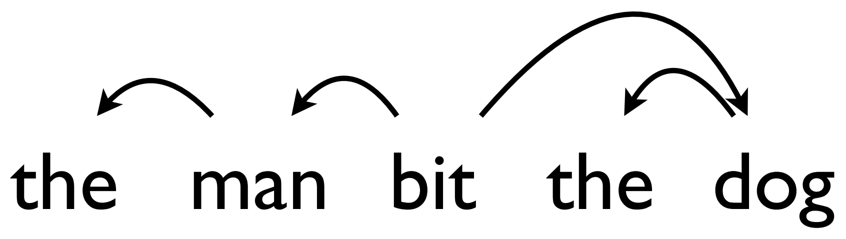
2018: linear-time
RNA structure prediction

Computational Linguistics => Computational Biology

linguistics

1955 Chomsky:
context-free grammars





computer science

1958 Backus & Naur:
CFGs in programming lang.

1960s CKY Parsing: **$O(n^3)$**

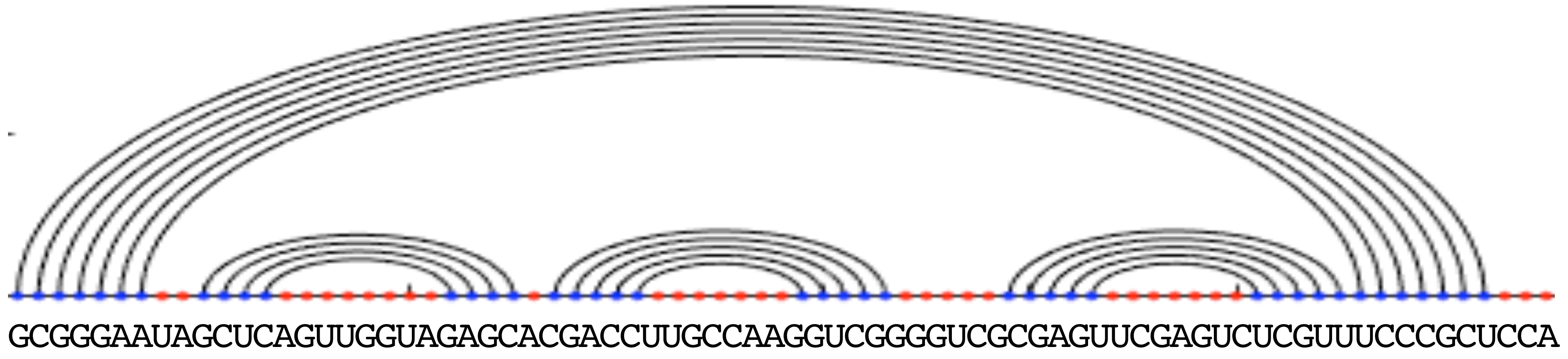
1965 Knuth: LR Parsing: **$O(n)$**

1986 Tomita: Generalized LR Parsing

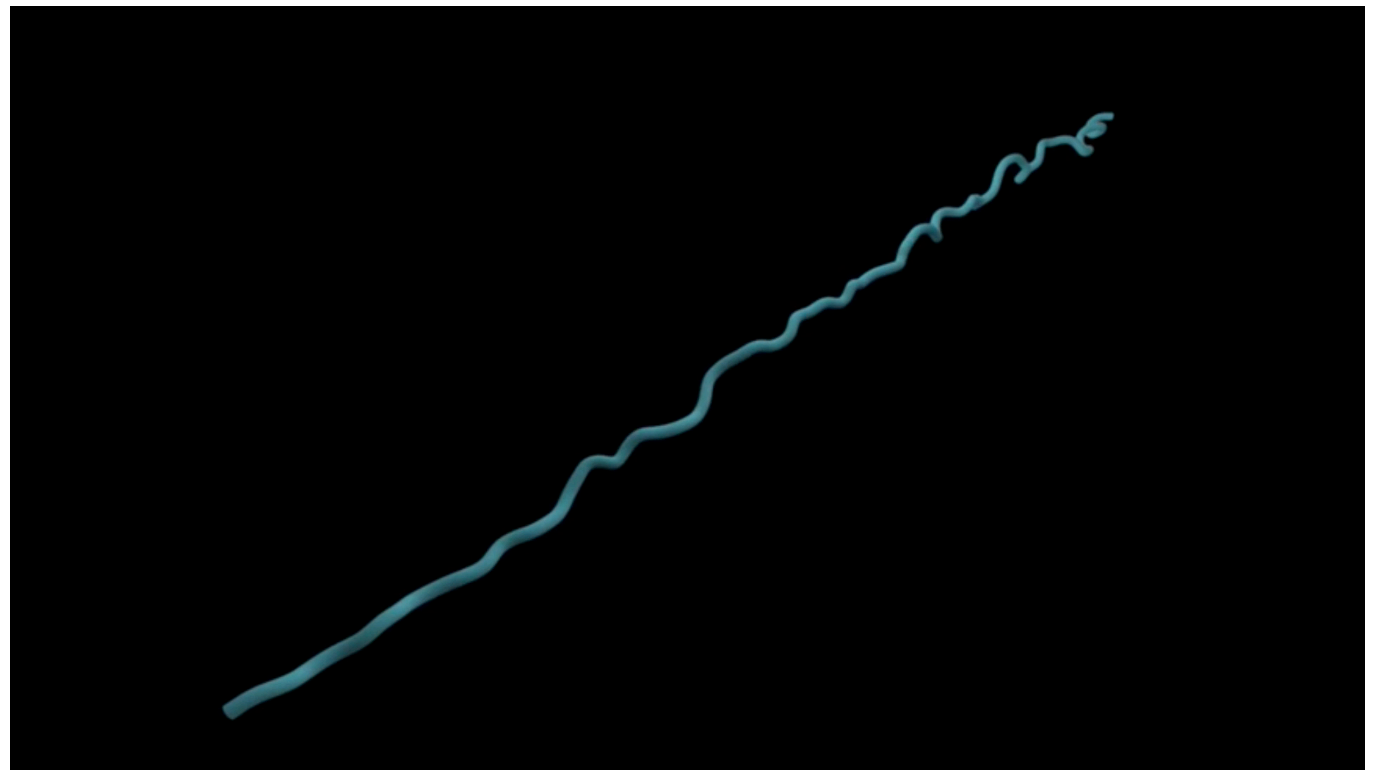
2010: linear-time DP parsing
(Huang & Sagae)

biology

1953 Watson & Crick: 
DNA double-helix



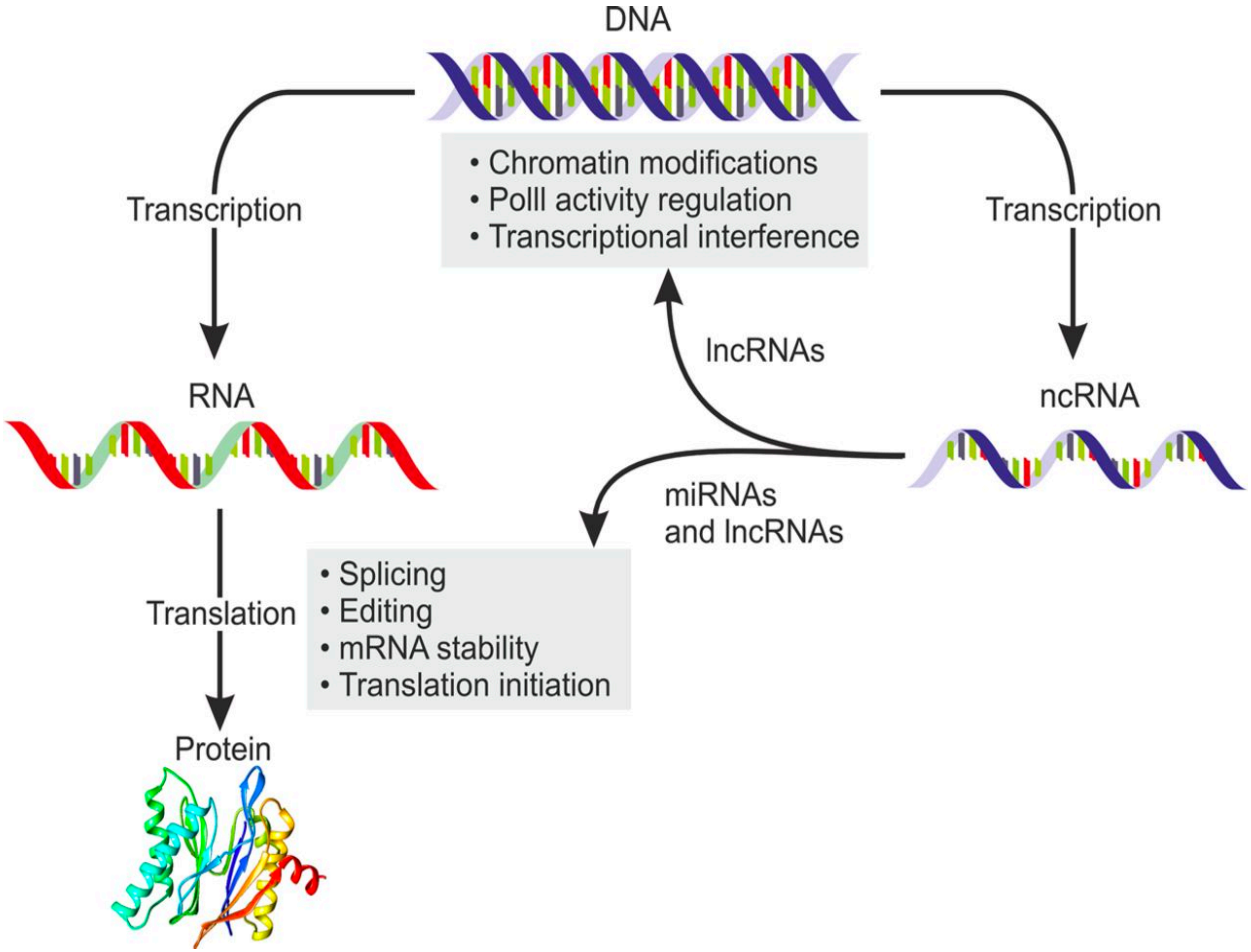
1980s: $O(n^3)$ CKY for RNA structures



2018: linear-time
RNA structure prediction

RNAs and Structure Prediction

RNA has dual roles:
informational (DNA=>RNA=>protein)
functional (non-coding RNAs)
knowing structures can infer function

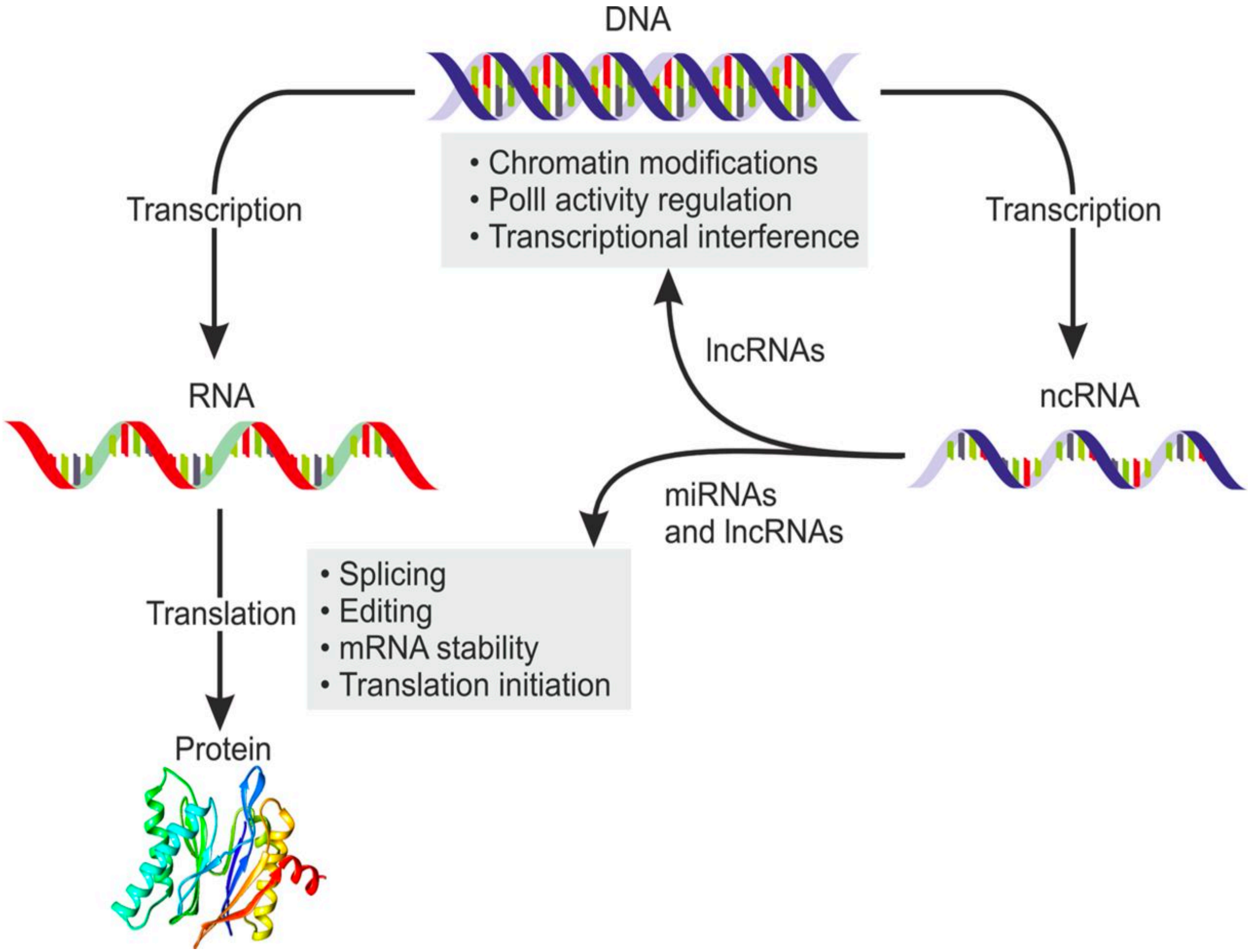


RNA sequence

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

RNAs and Structure Prediction

RNA has dual roles:
informational (DNA=>RNA=>protein)
functional (non-coding RNAs)
knowing structures can infer function

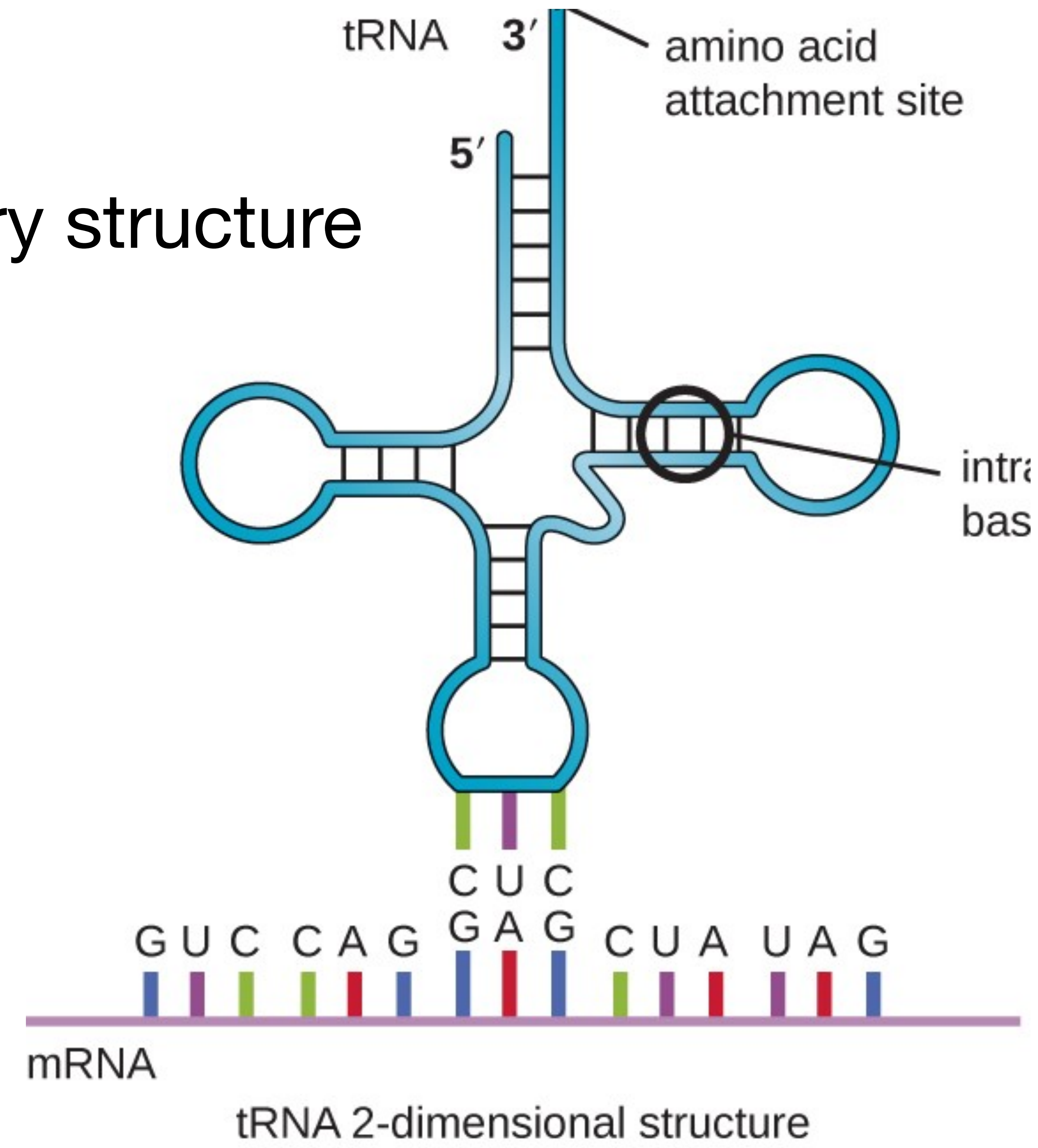


RNA sequence

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

structure prediction
("RNA folding")

secondary structure



RNA Secondary Structure Prediction

allowed pairs: G-C A-U G-U

assume no crossing pairs

example: transfer RNA (tRNA)

input x GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

RNA Secondary Structure Prediction

allowed pairs: G-C A-U G-U

assume no crossing pairs

example: transfer RNA (tRNA)

input x GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

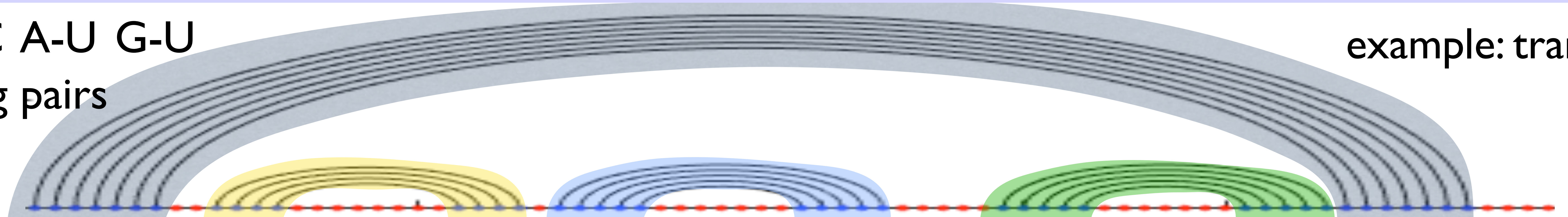
output y (((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

RNA Secondary Structure Prediction

allowed pairs: G-C A-U G-U

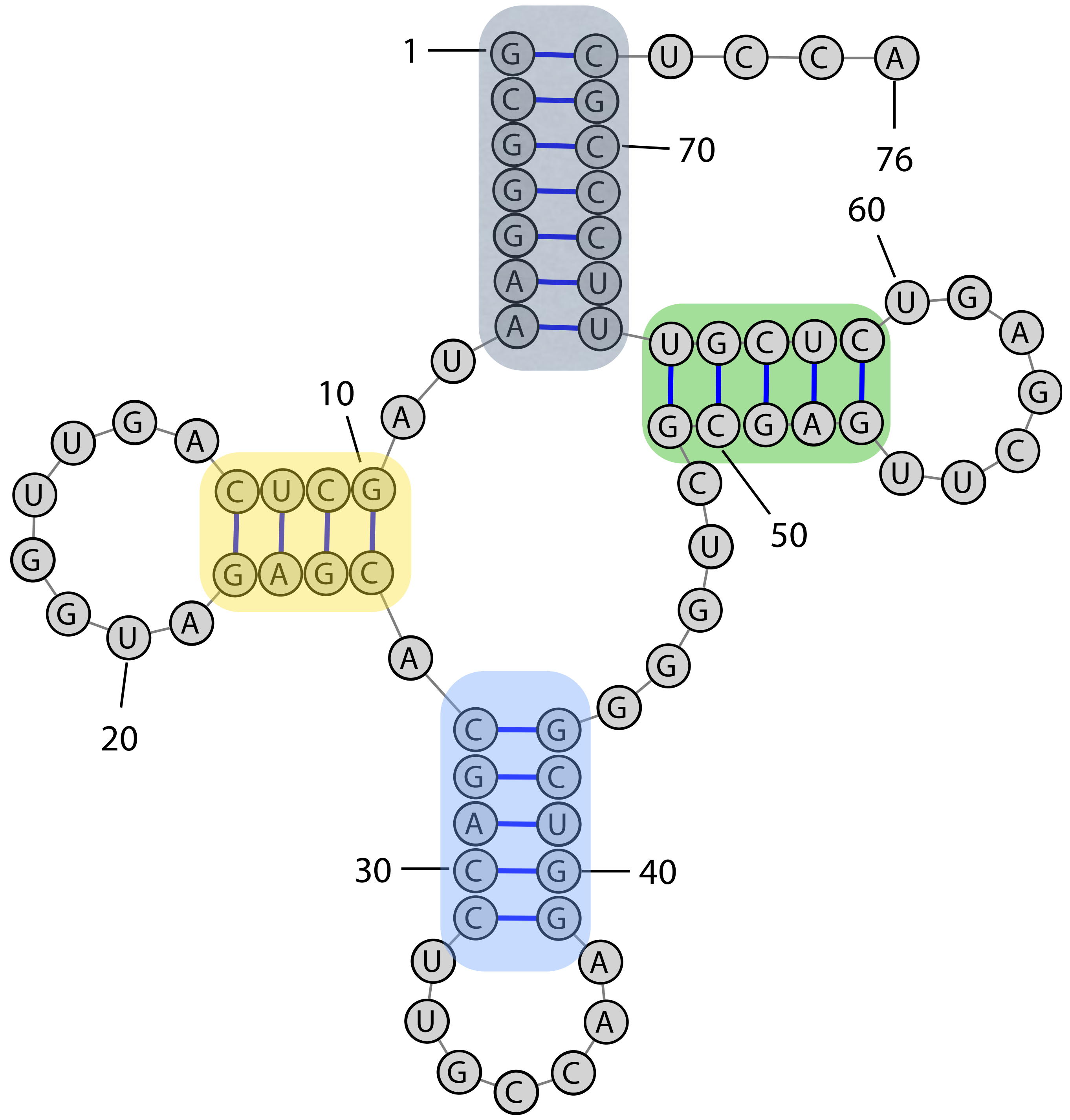
assume no crossing pairs

example: transfer RNA (tRNA)



input x GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

output y (((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

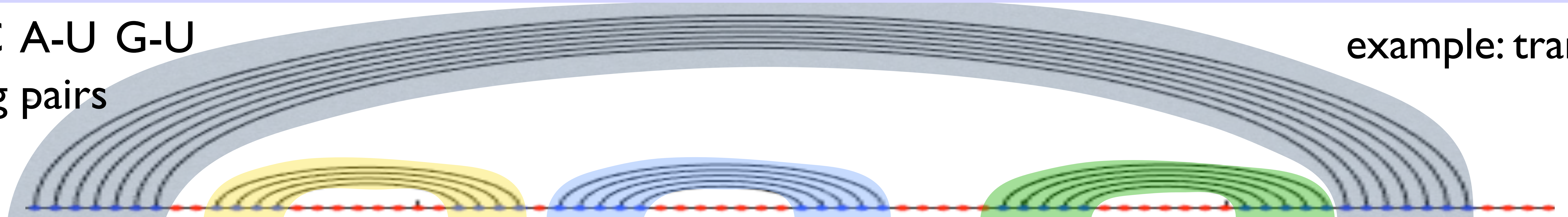


RNA Secondary Structure Prediction

allowed pairs: G-C A-U G-U

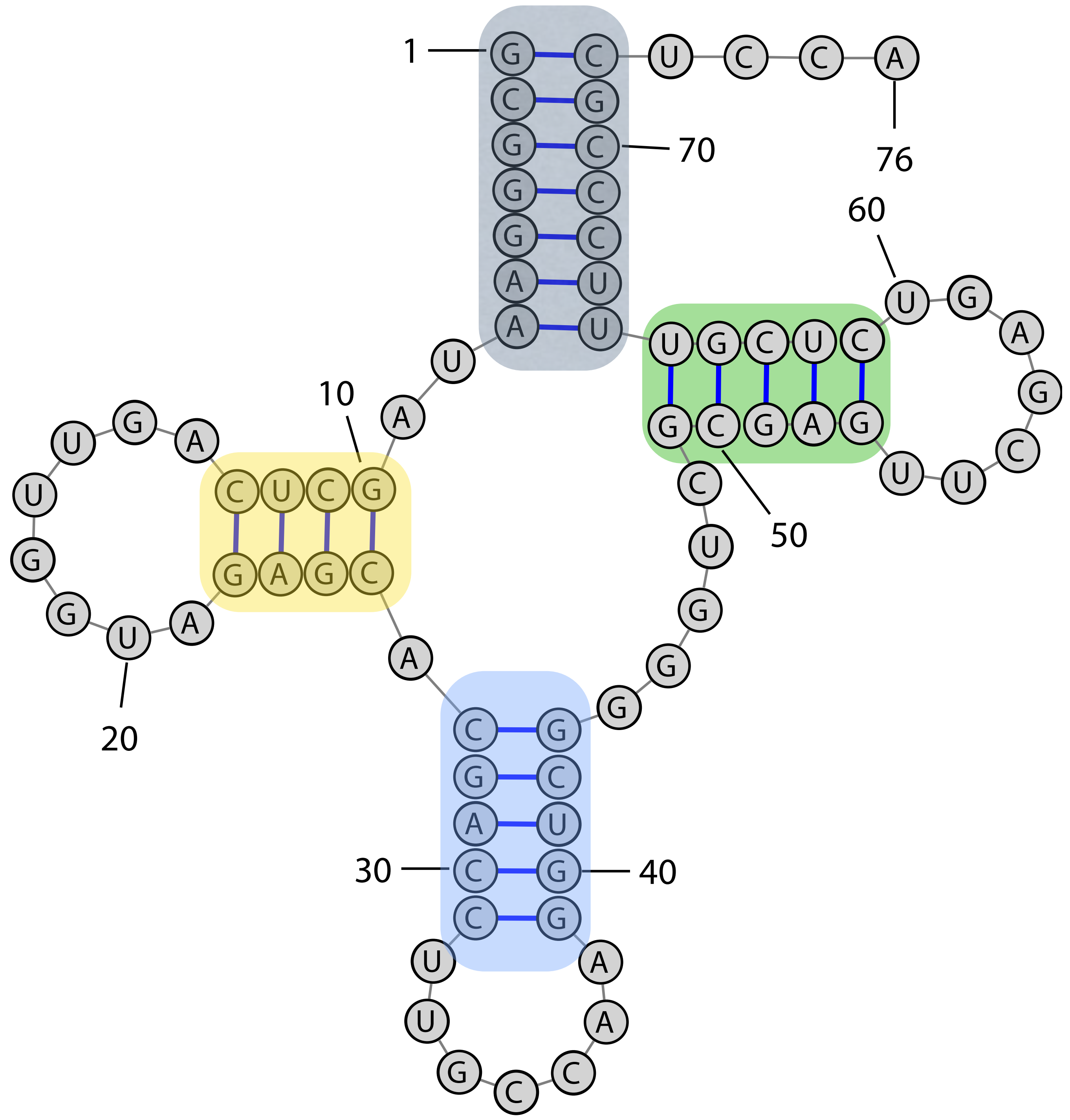
assume no crossing pairs

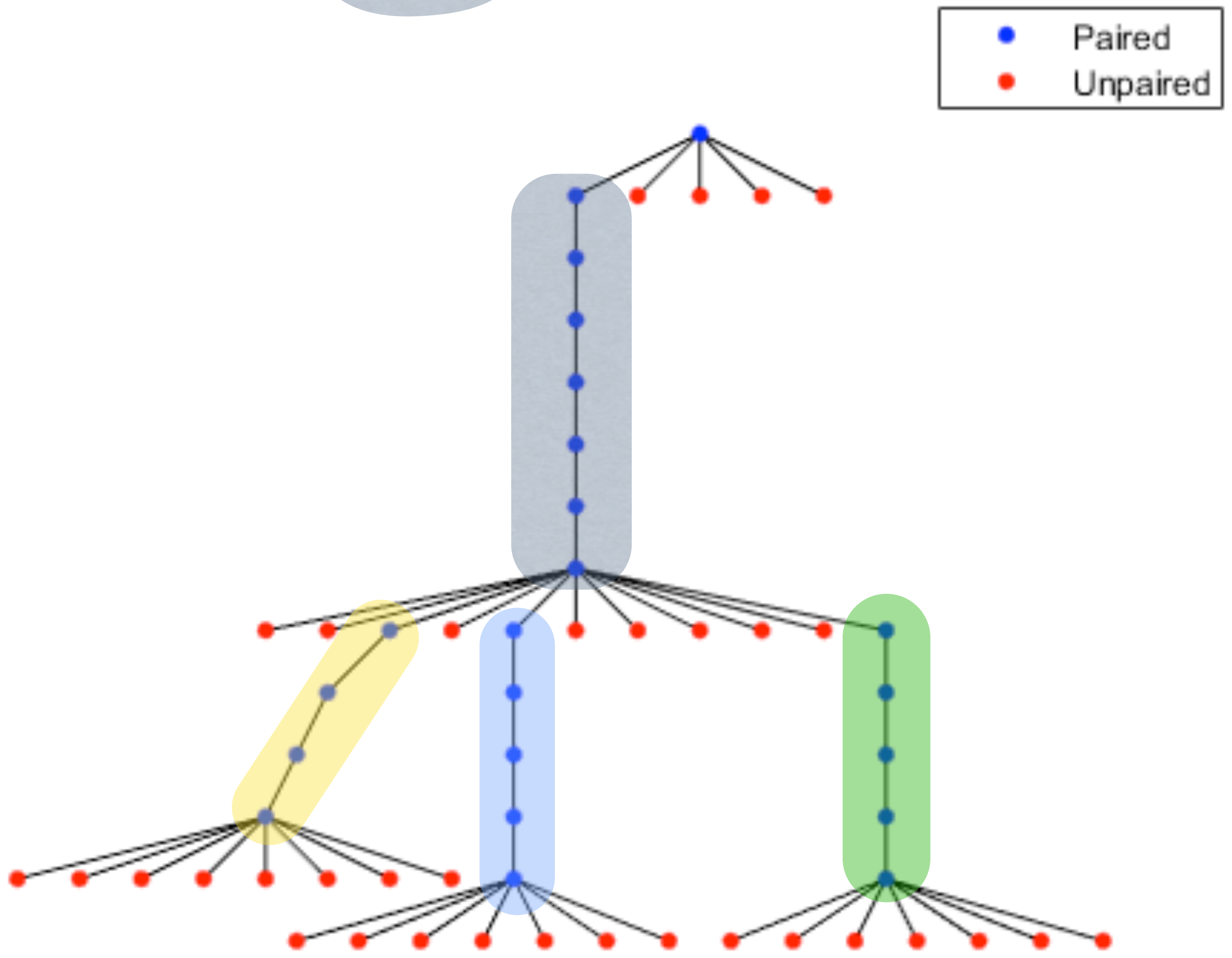
example: transfer RNA (tRNA)



input x GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

output y (((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....



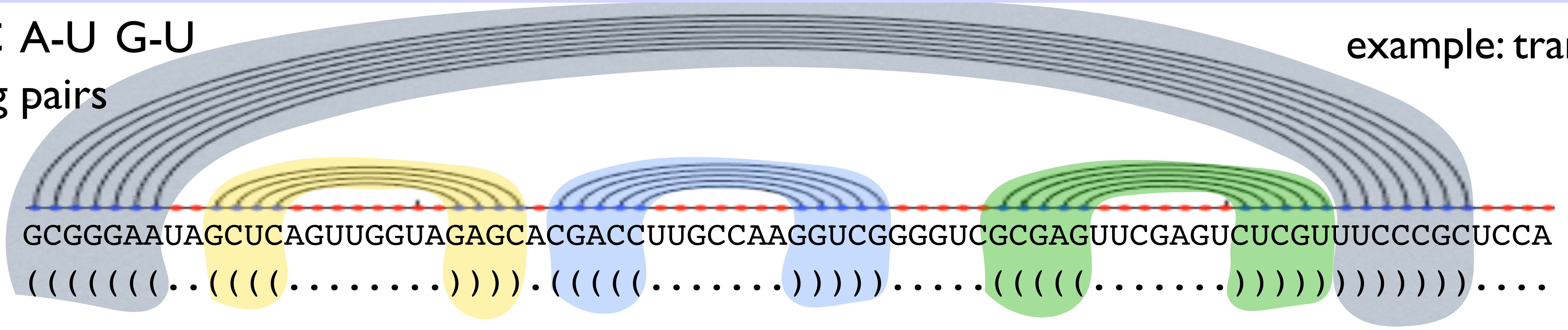


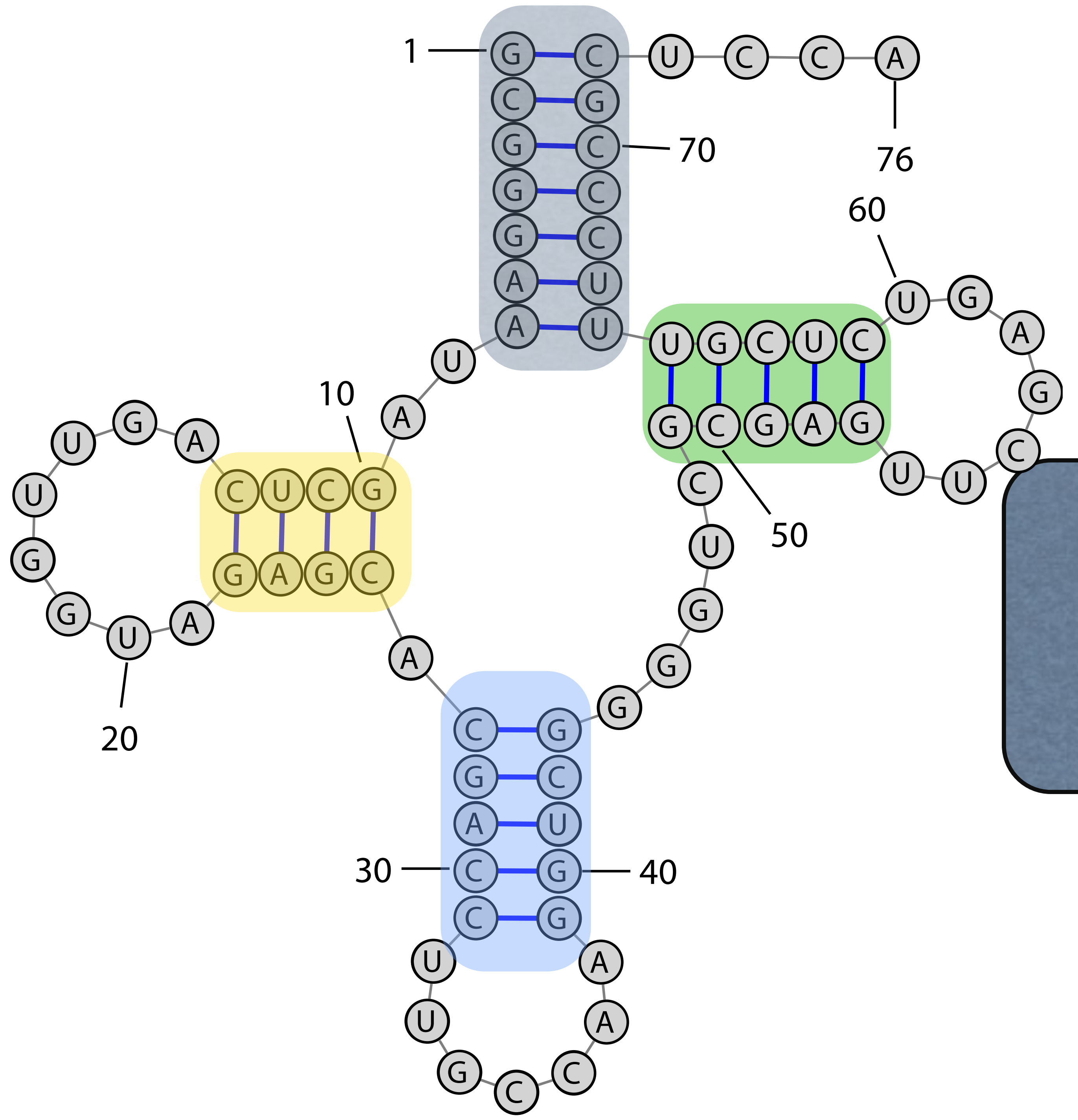
parse tree

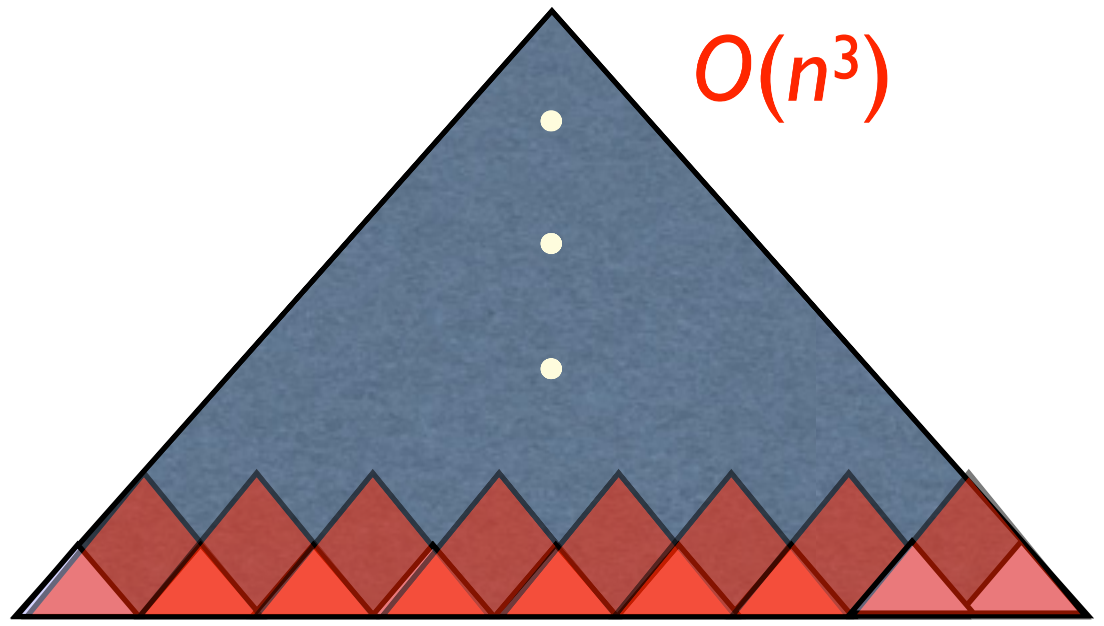
RNA Secondary Structure Prediction

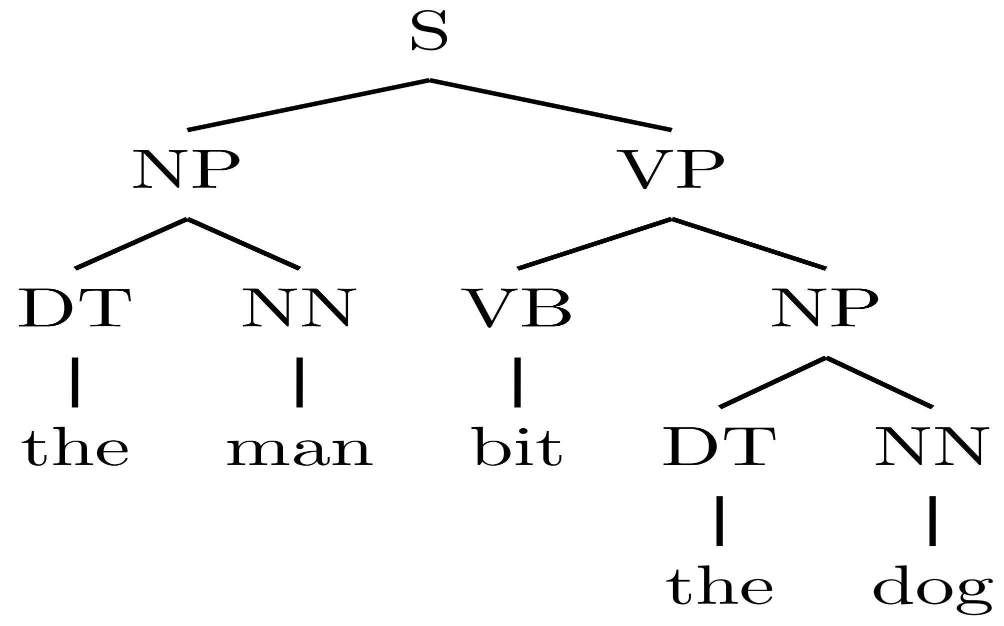
allowed pairs: G-C A-U G-U
assume no crossing pairs

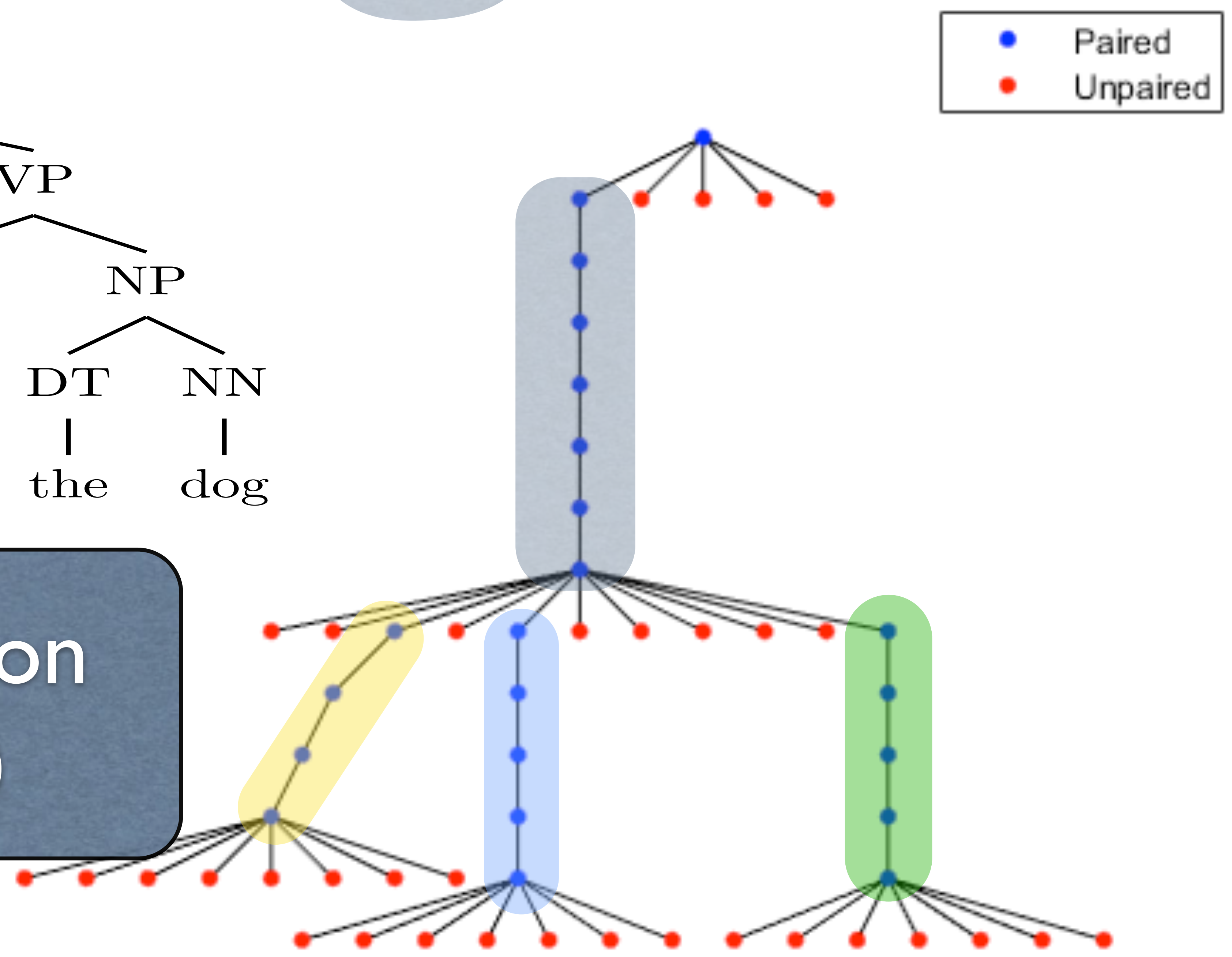
example: transfer RNA (tRNA)











parse tree

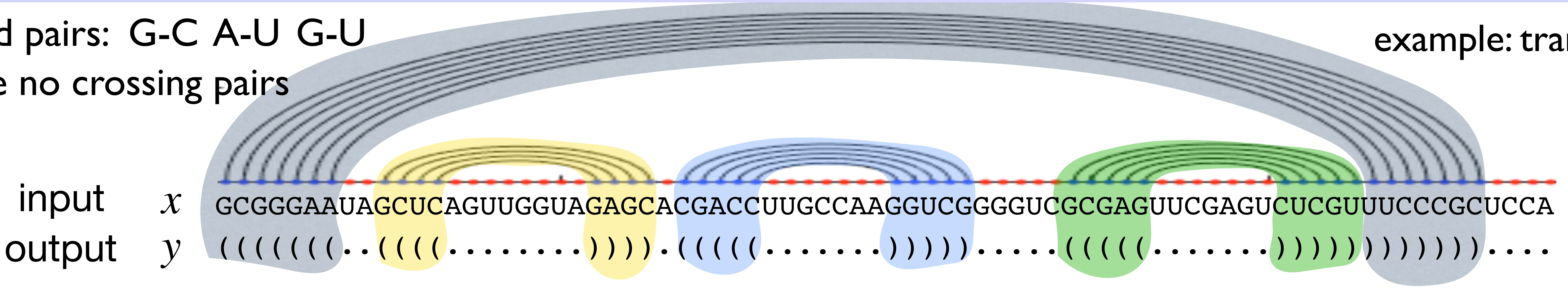
problem: standard structure prediction algorithms are way too slow: $O(n^3)$

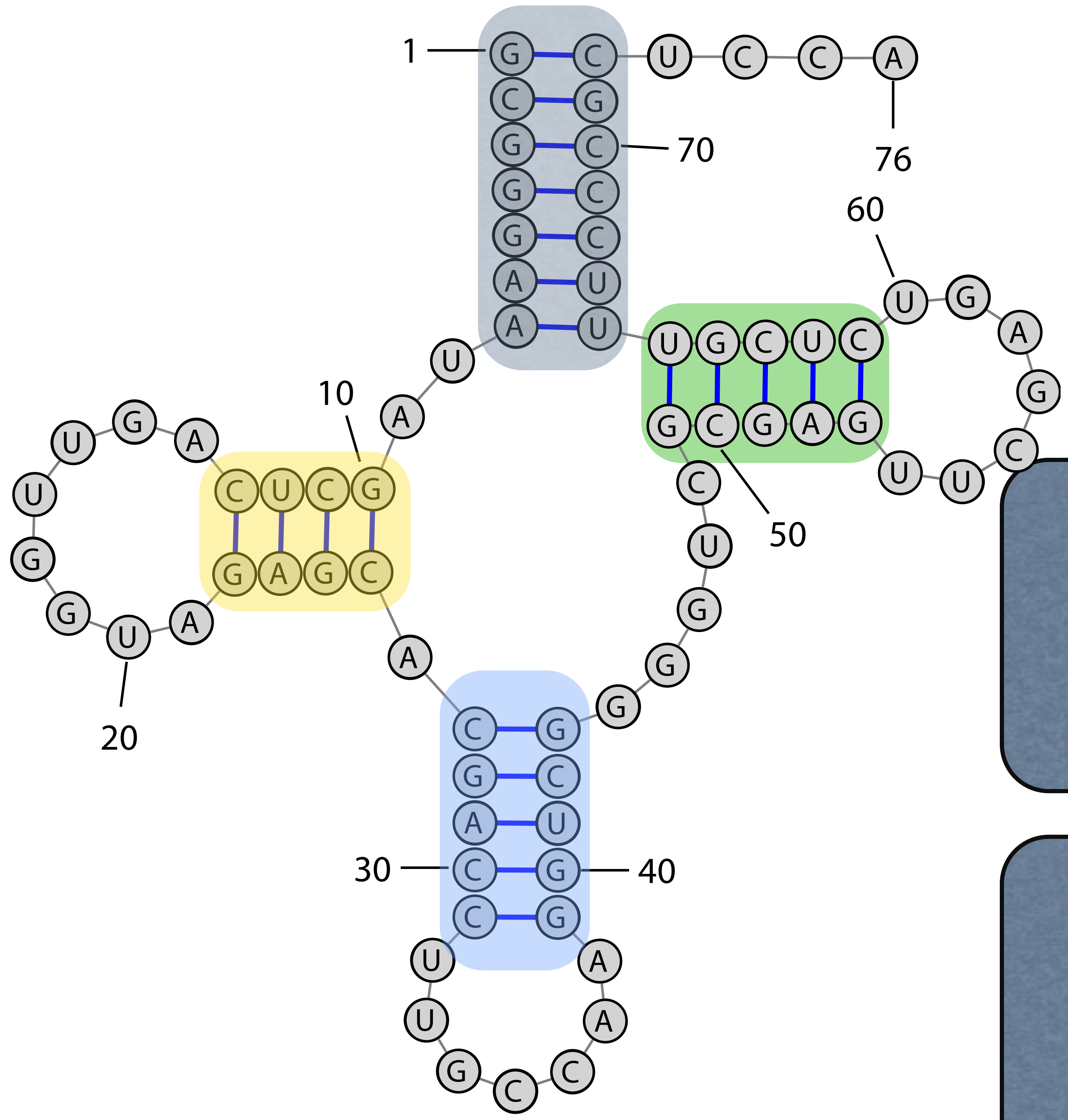
RNA Secondary Structure Prediction

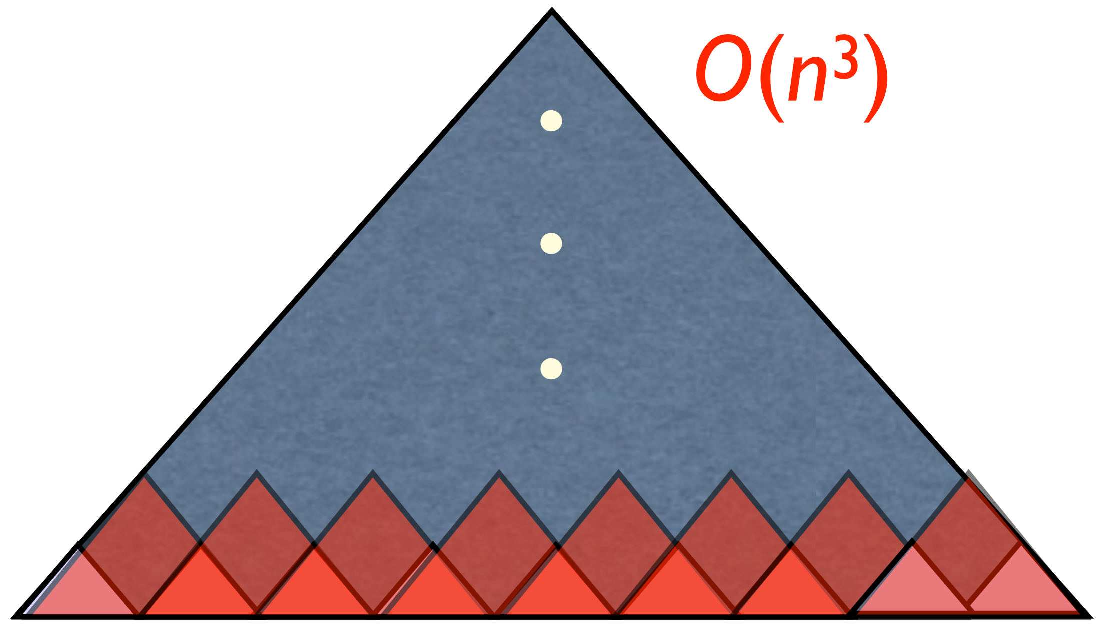
allowed pairs: G-C A-U G-U

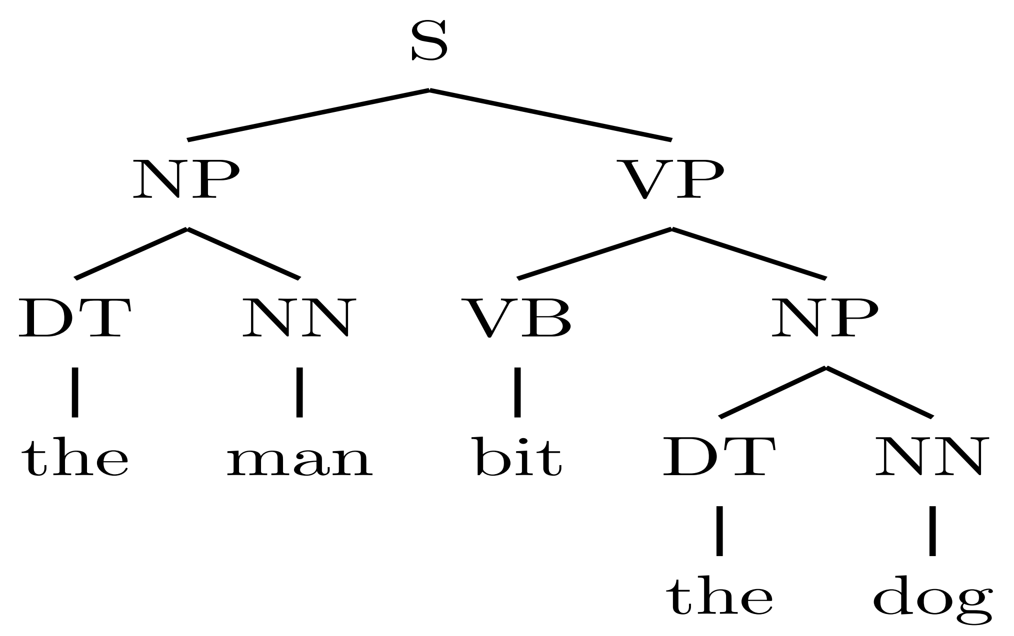
assume no crossing pairs

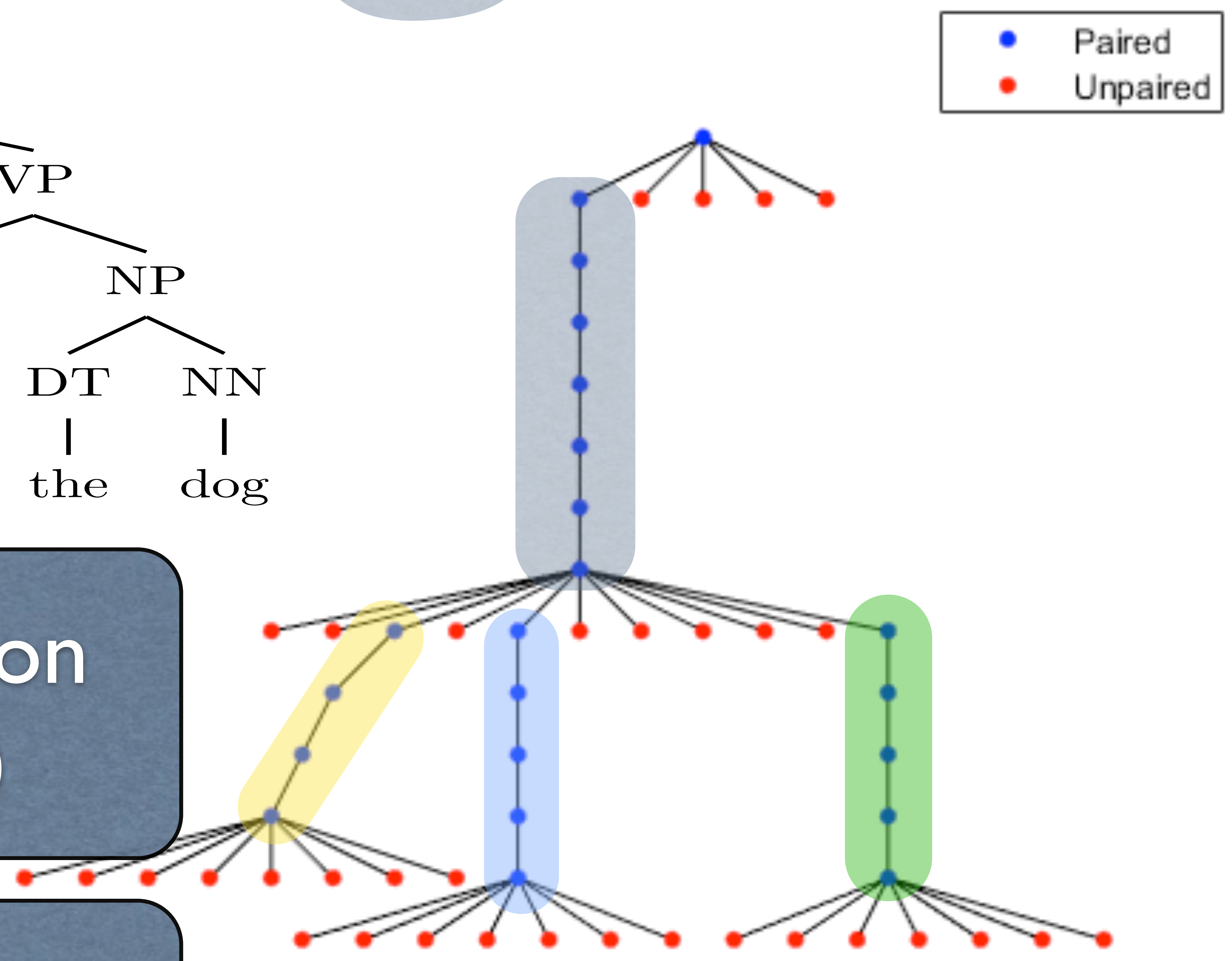
example: transfer RNA (tRNA)











parse tree

problem: standard structure prediction algorithms are way too slow: $O(n^3)$

solution: adapt my linear-time dynamic programming algorithms from parsing

How to Fold RNAs in Linear-Time?

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

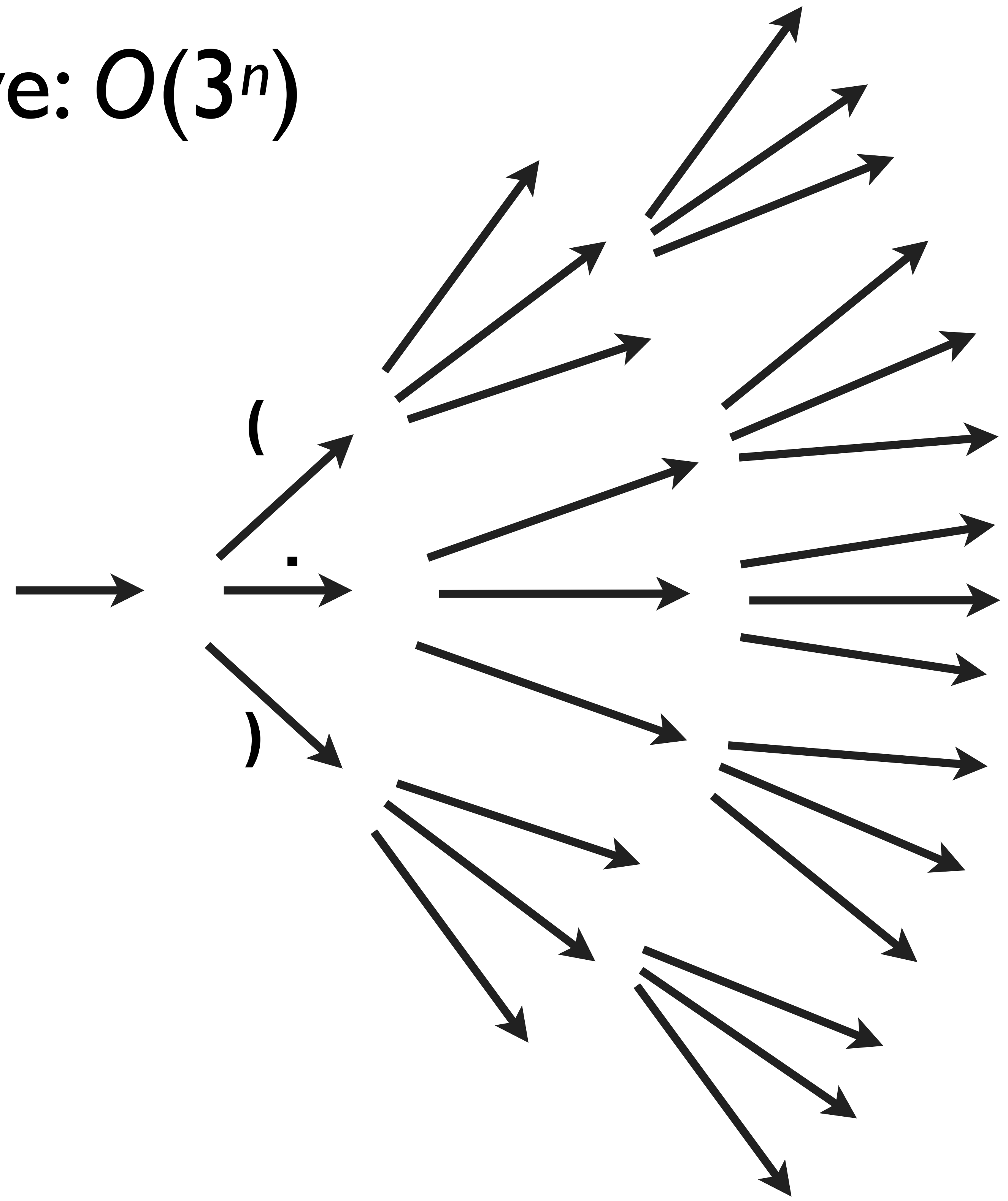
- idea 0: tag each nucleotide from left to right
  - maintain a stack: push “(”, pop “)”, skip “.”
  - exhaustive: $O(3^n)$

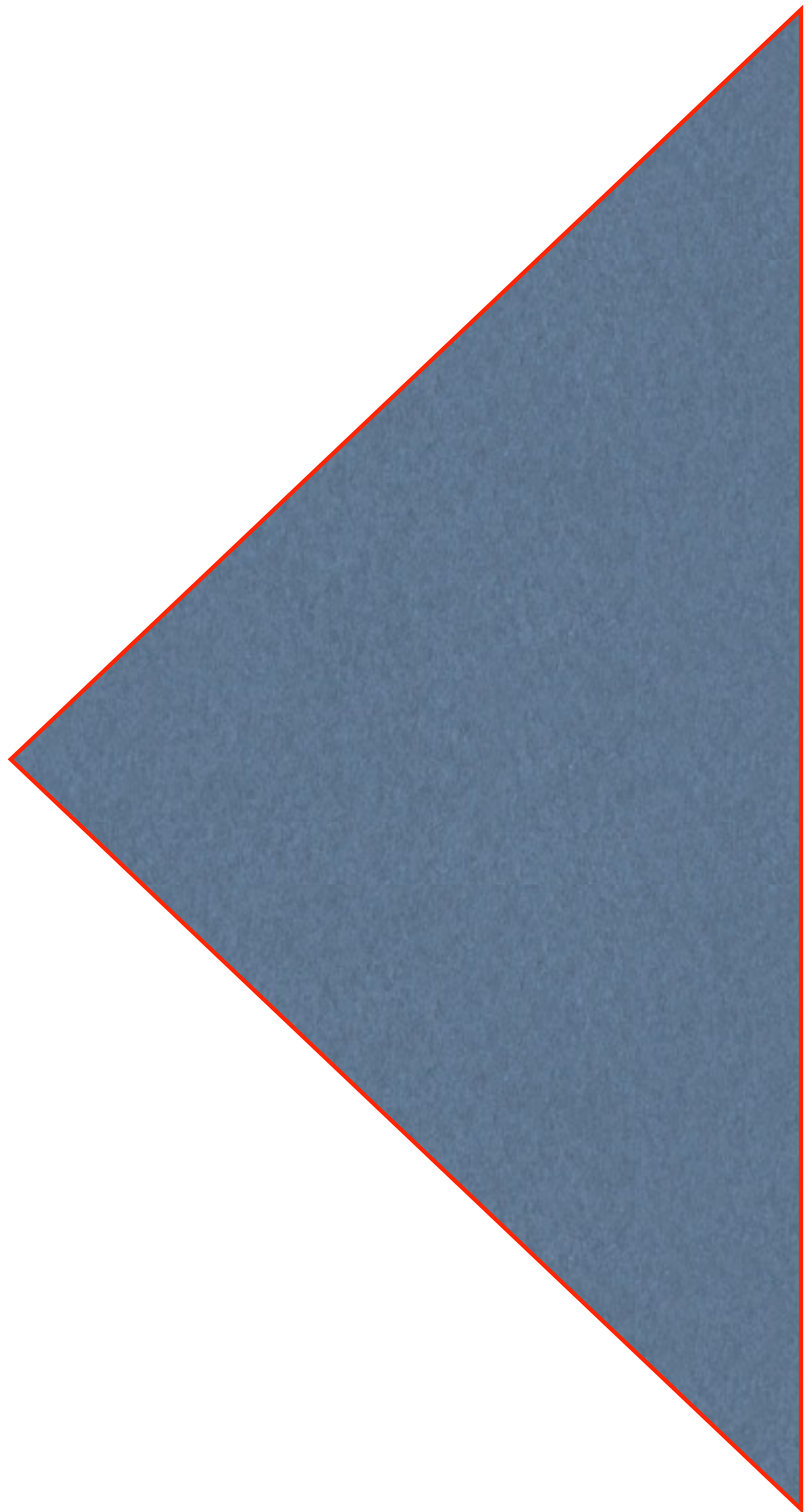
How to Fold RNAs in Linear-Time?

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

- idea 0: tag each nucleotide from left to right
  - maintain a stack: push "(", pop ")", skip "."
  - exhaustive: $O(3^n)$



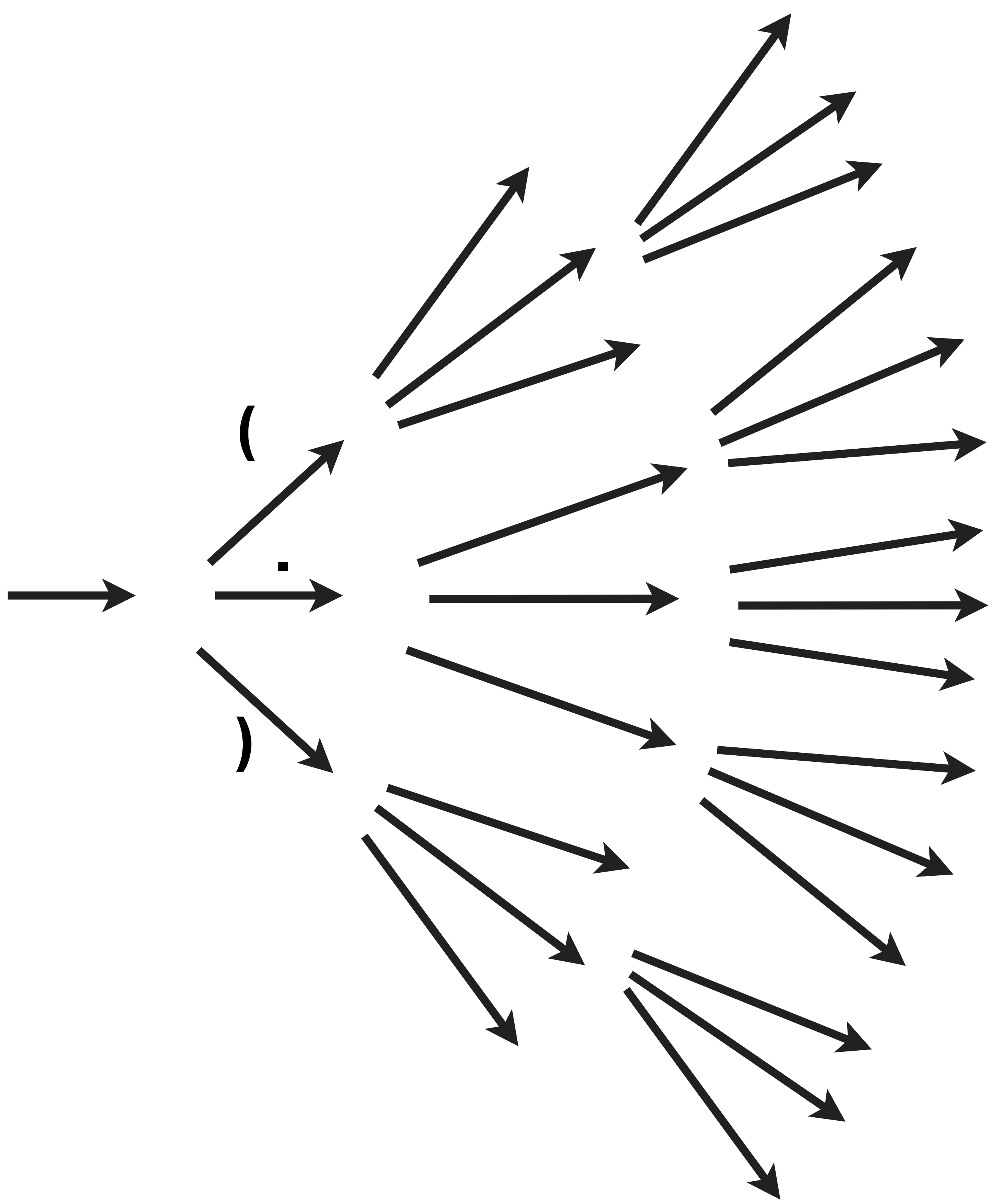


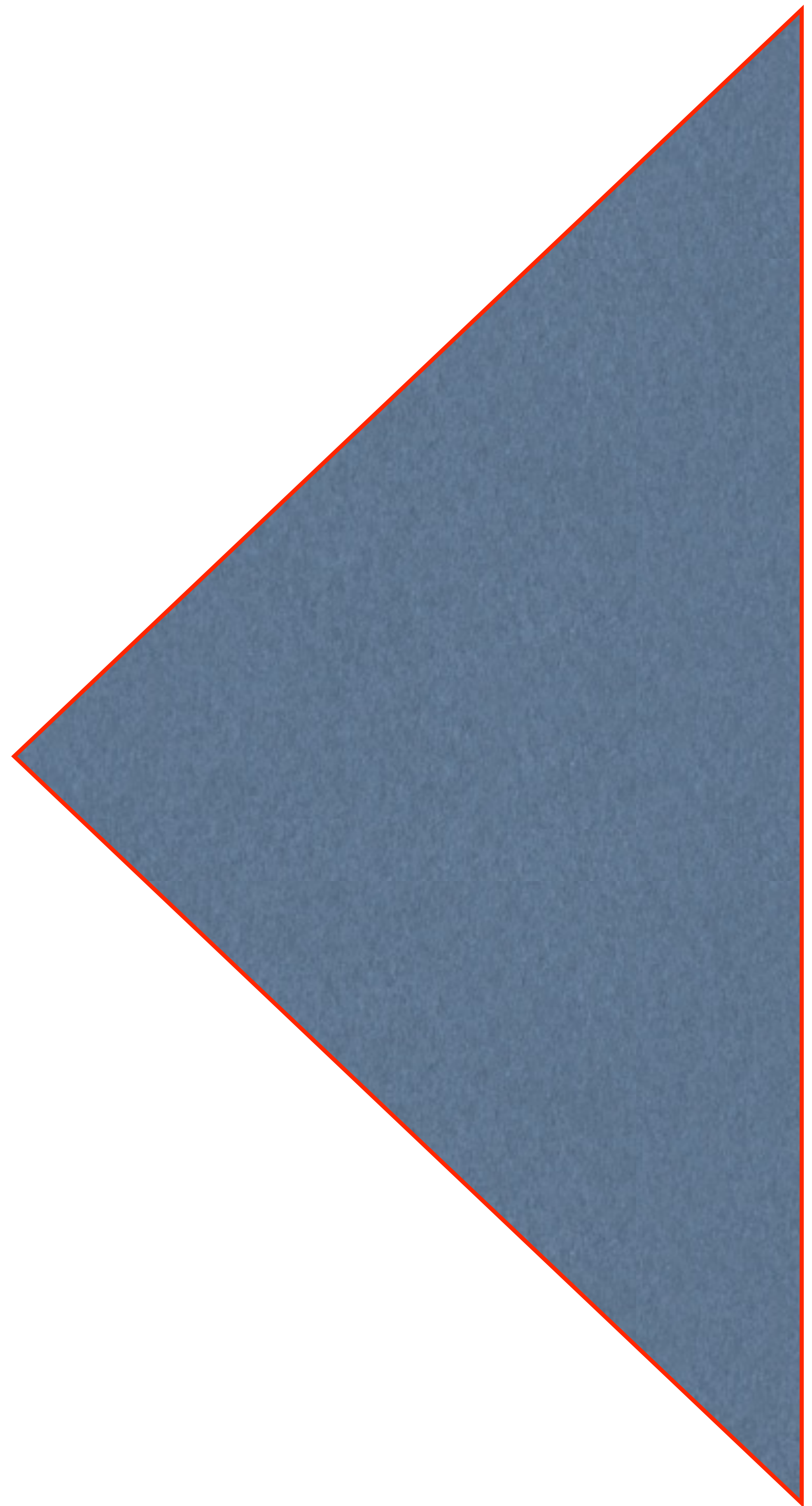
How to Fold RNAs in Linear-Time?

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

- idea 1: DP by merging "equivalent states"
  - maintain graph-structured stacks
  - DP: $O(n^3)$

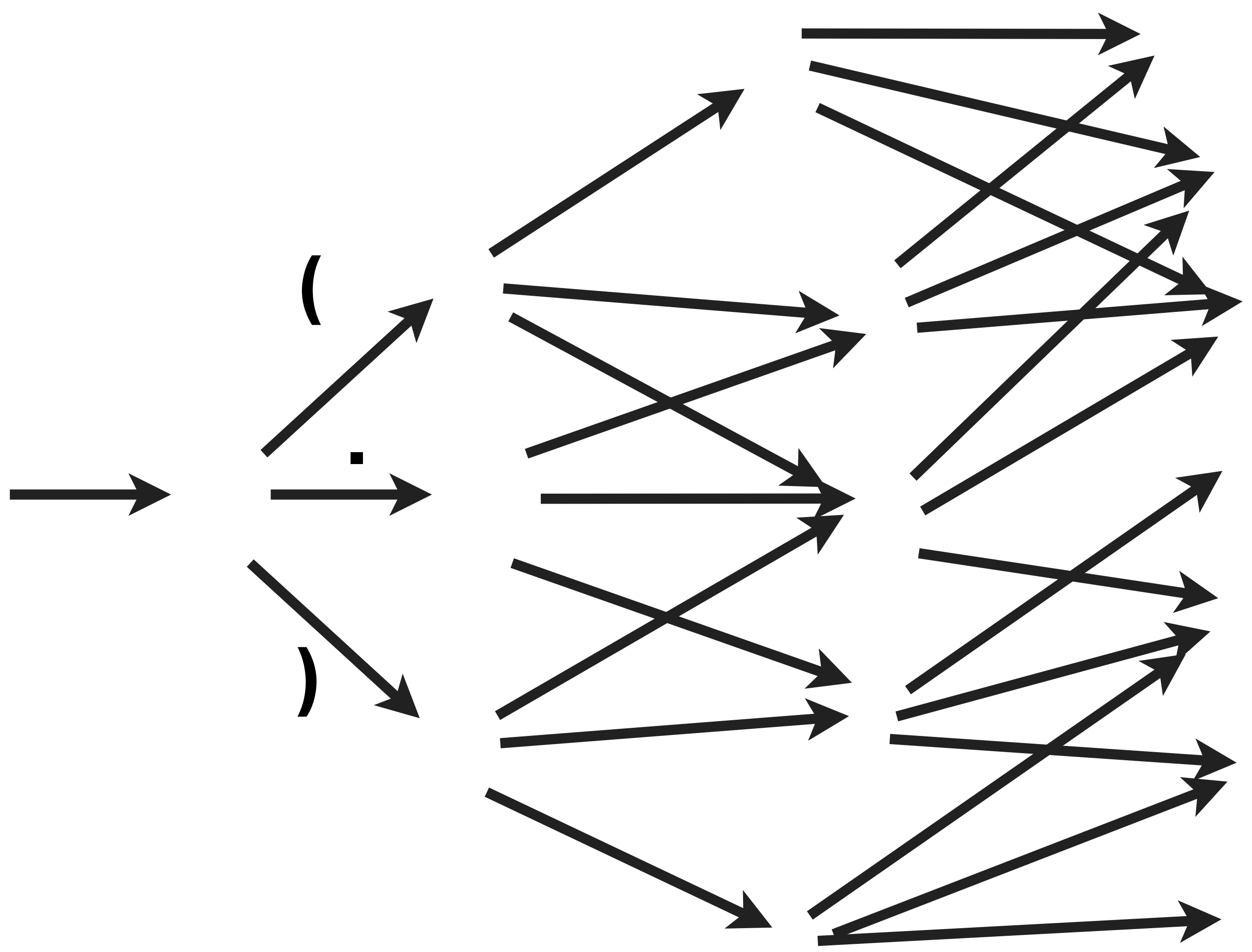


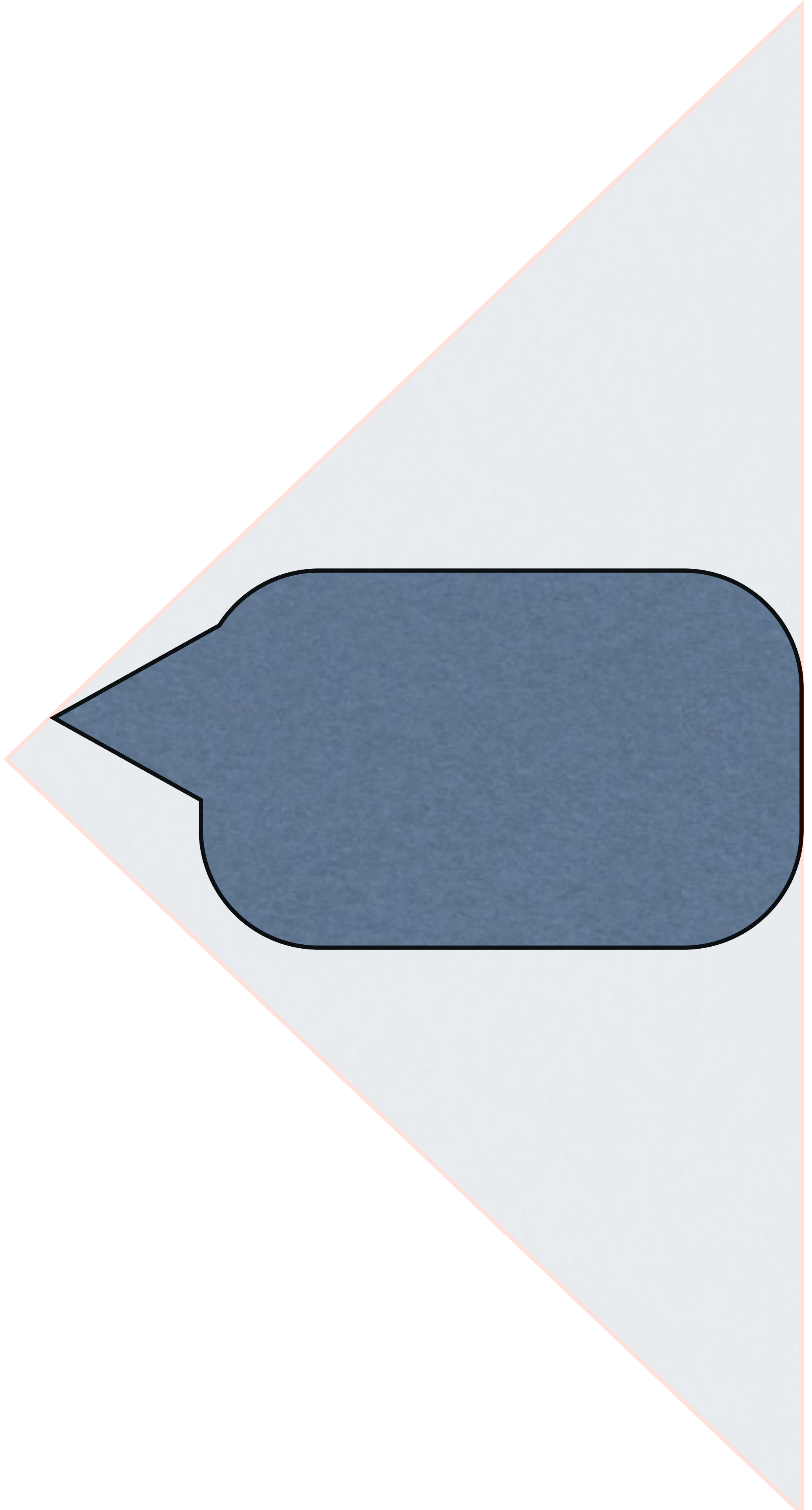


How to Fold RNAs in Linear-Time?

```
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA
(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....
```

- idea 1: DP by merging “equivalent states”
  - maintain graph-structured stacks
  - DP: $O(n^3)$

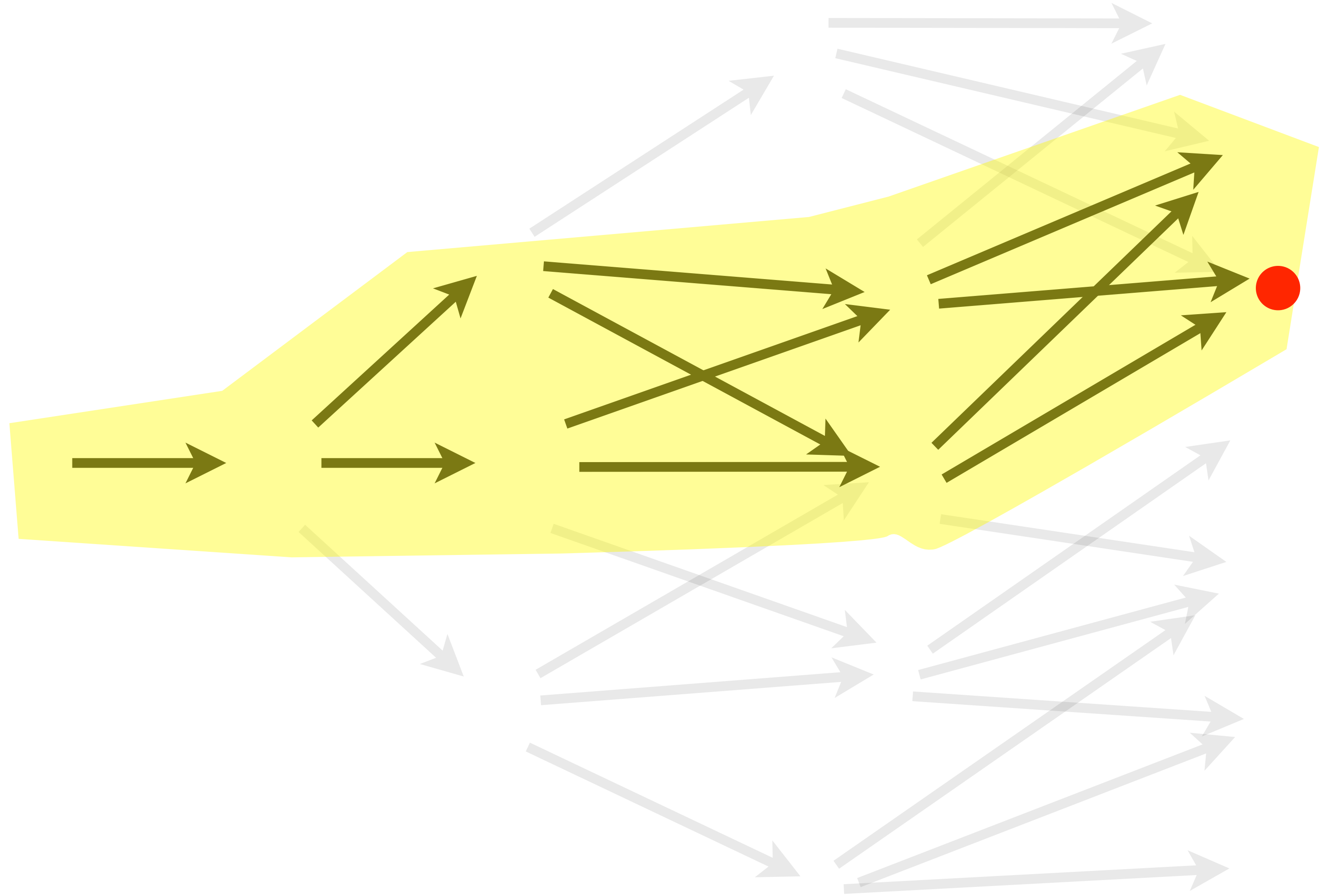


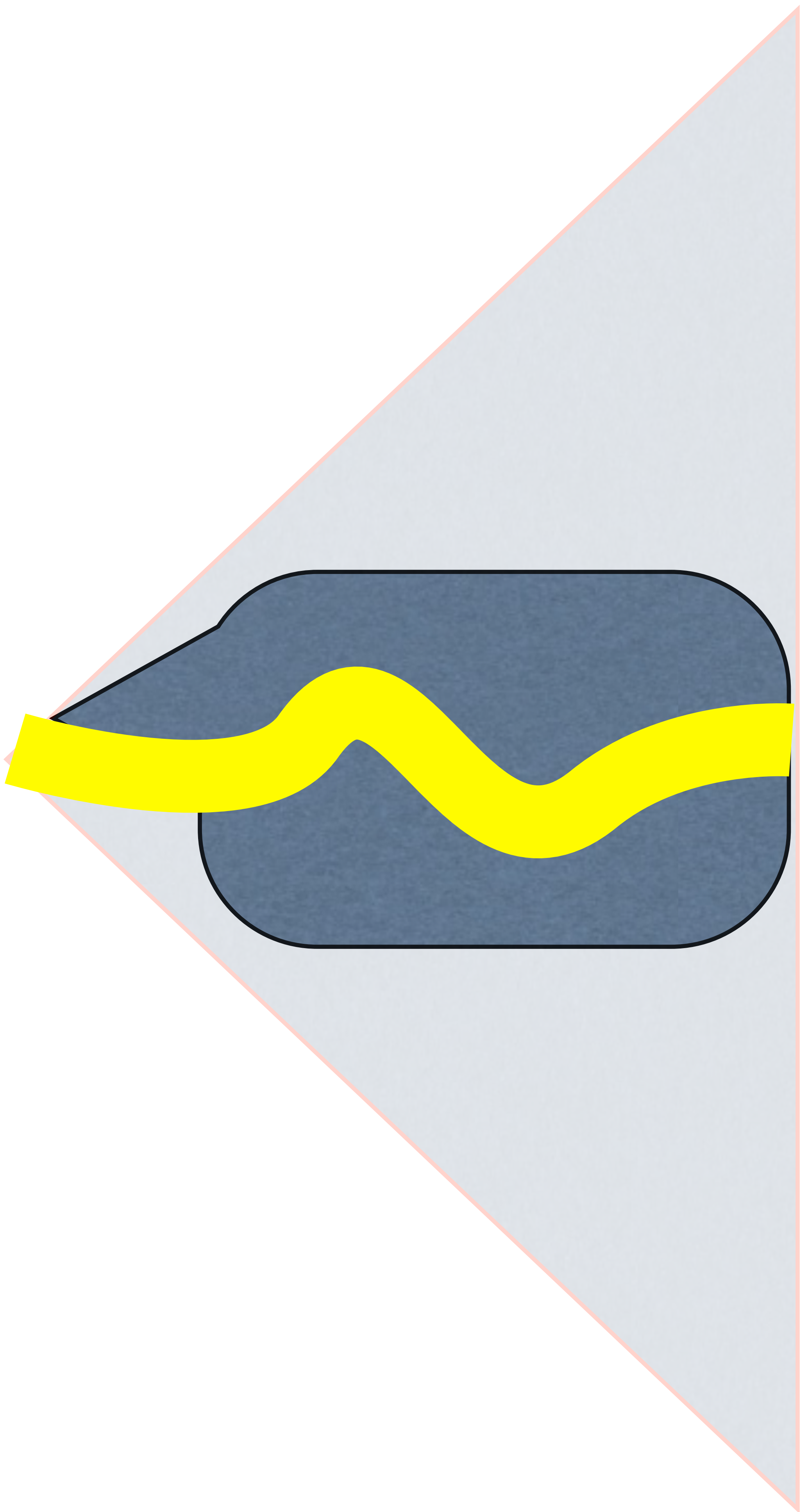


How to Fold RNAs in Linear-Time?

```
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA
(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....
```

- idea 2: approximate search: beam pruning
  - keep only top *b* states per step
  - DP+beam: $O(n)$



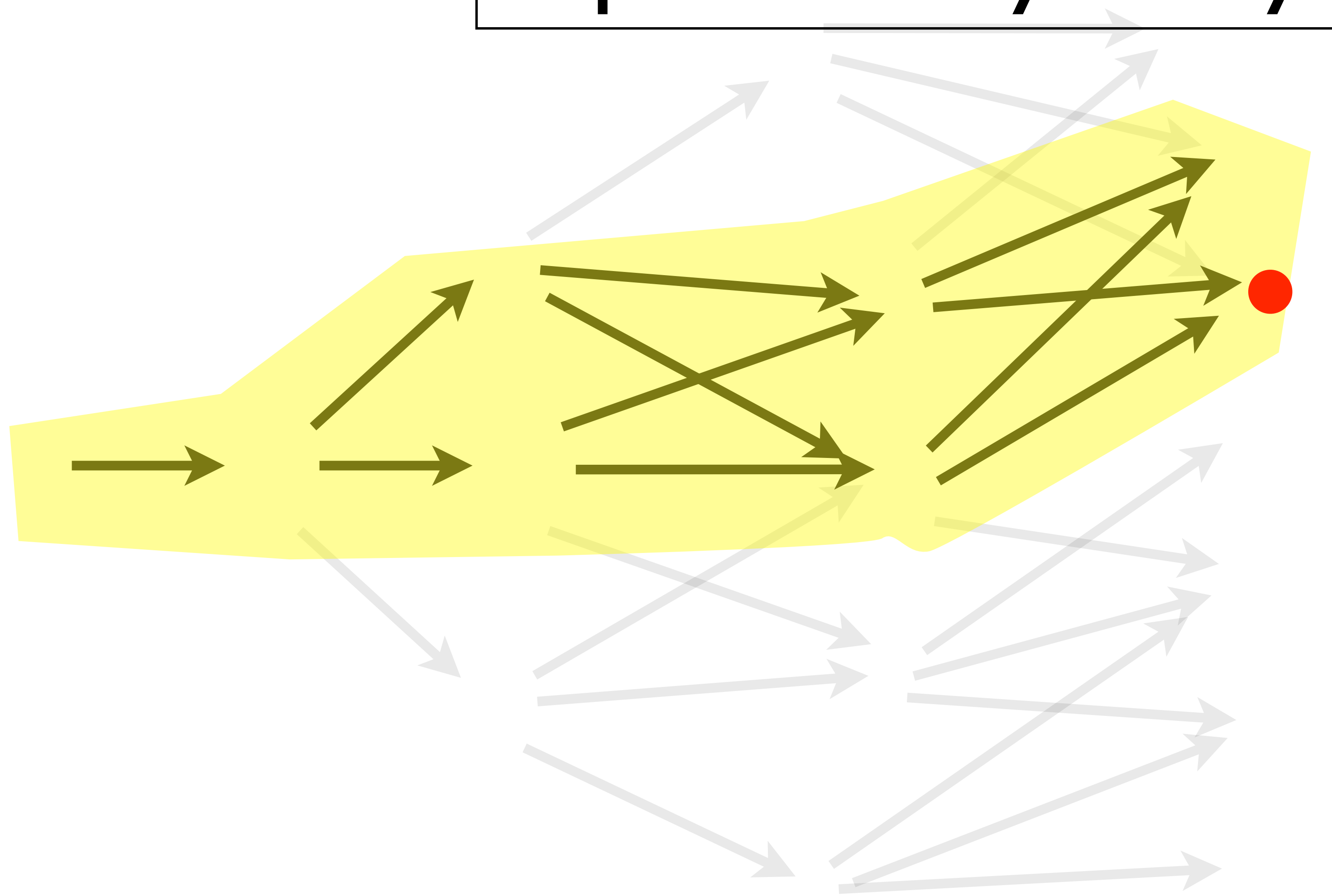


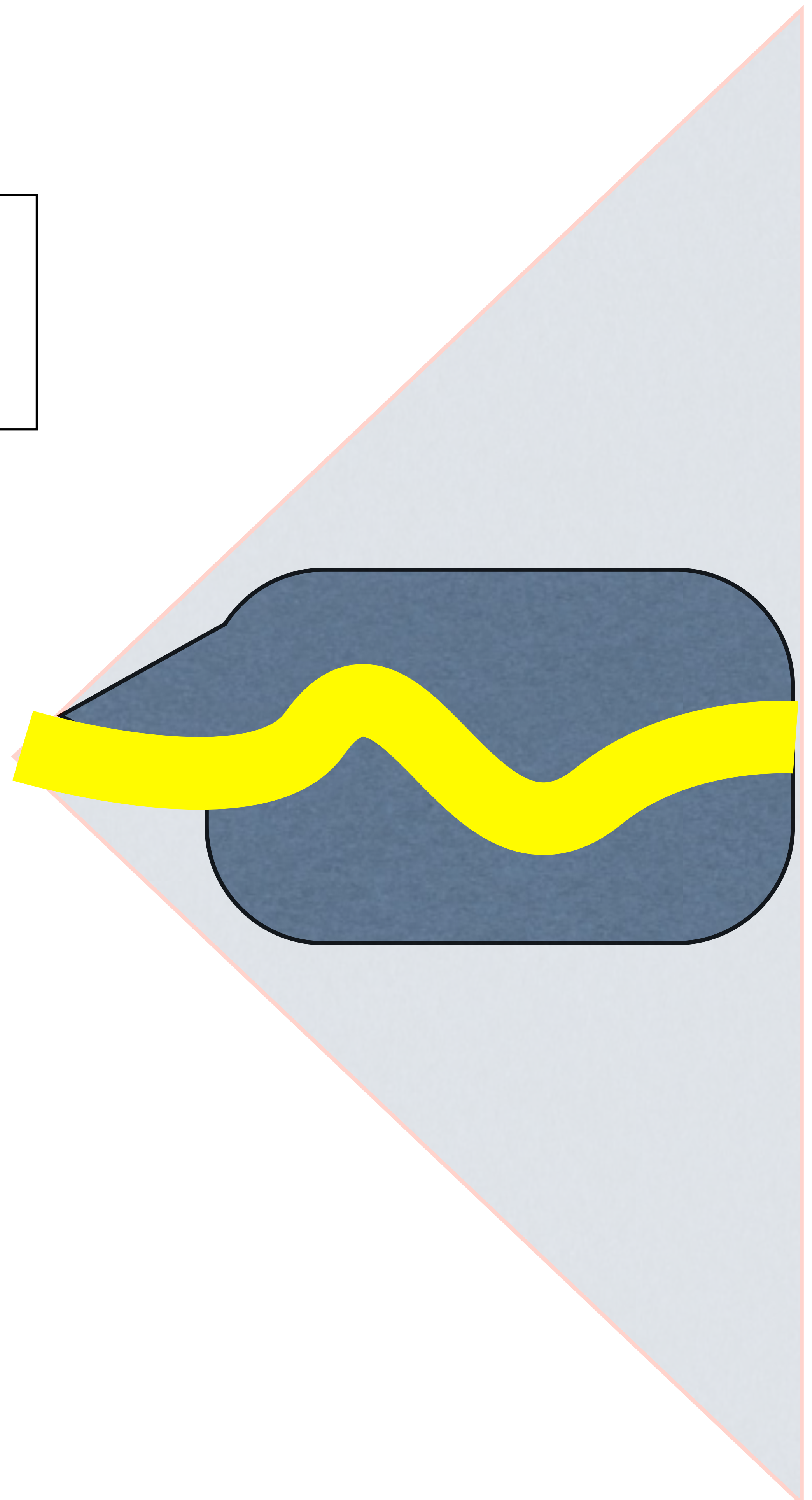
How to Fold RNAs in Linear-Time?

```
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCGCUCCA
(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....
```

- idea 2: approximate search: beam pruning
  - keep only top *b* states per step
  - DP+beam: $O(n)$

each DP state corresponds to exponentially many non-DP states





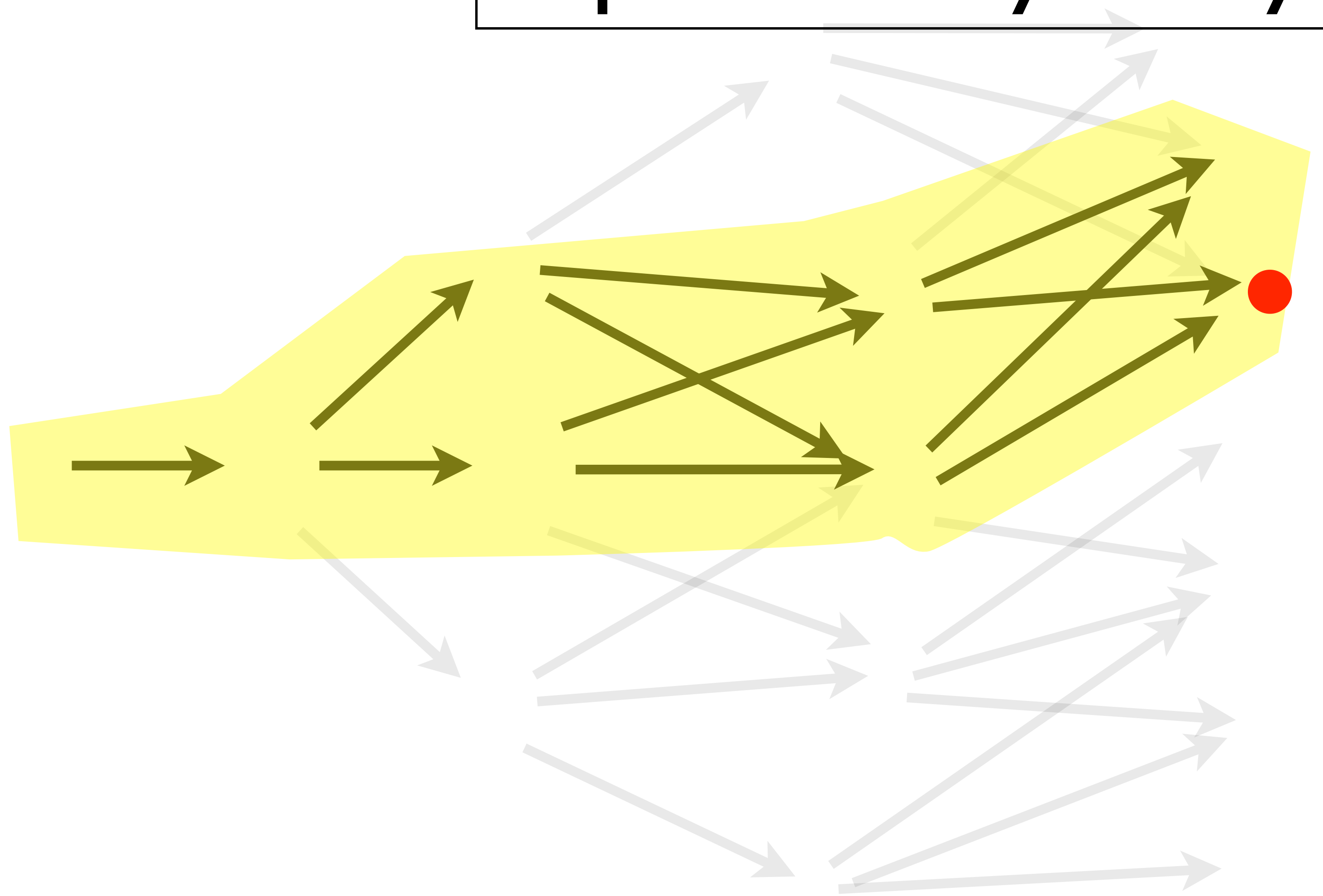
How to Fold RNAs in Linear-Time?

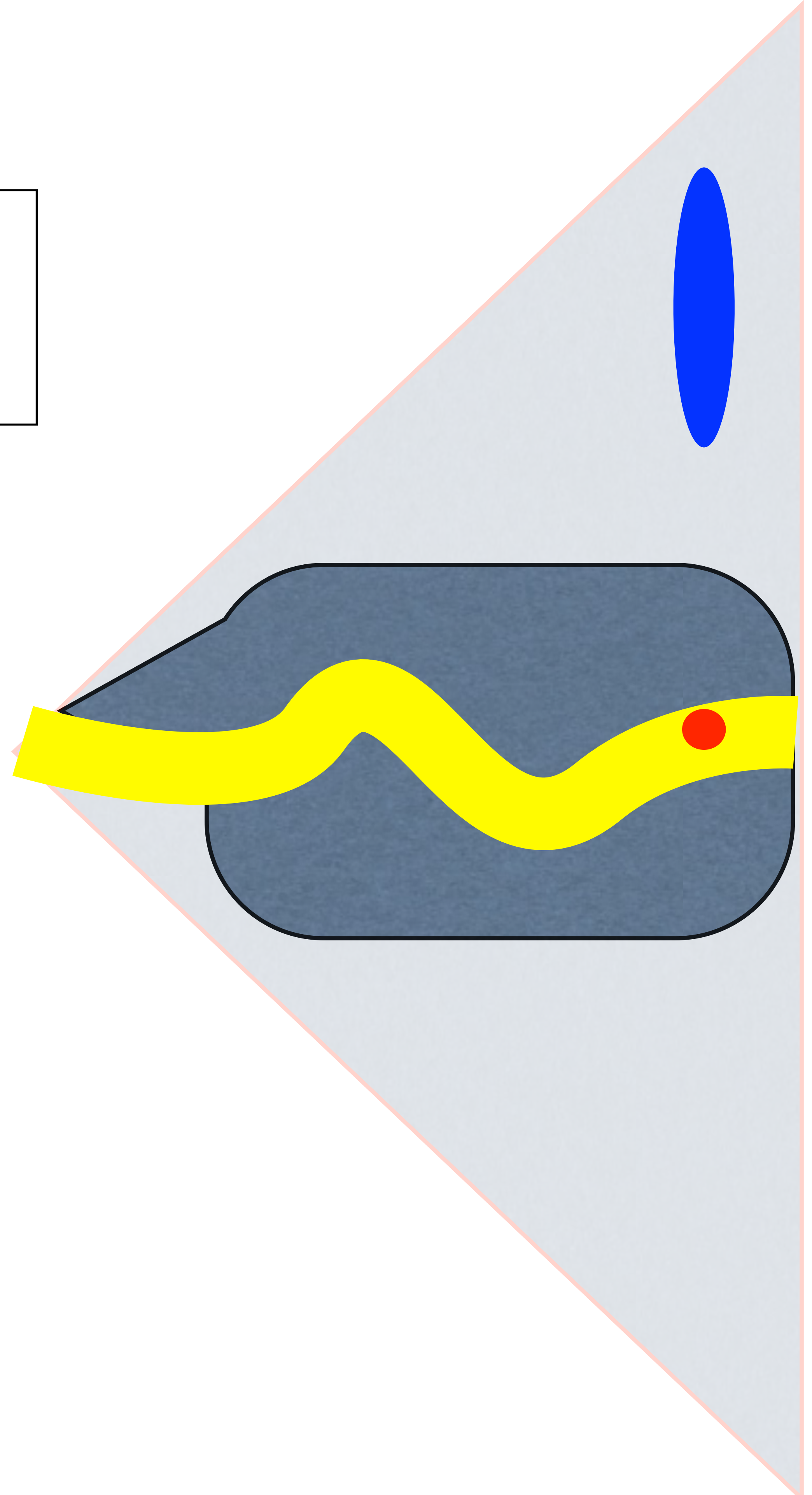
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

- idea 2: approximate search: beam pruning
  - keep only top *b* states per step
  - DP+beam: $O(n)$

each DP state corresponds to exponentially many non-DP states





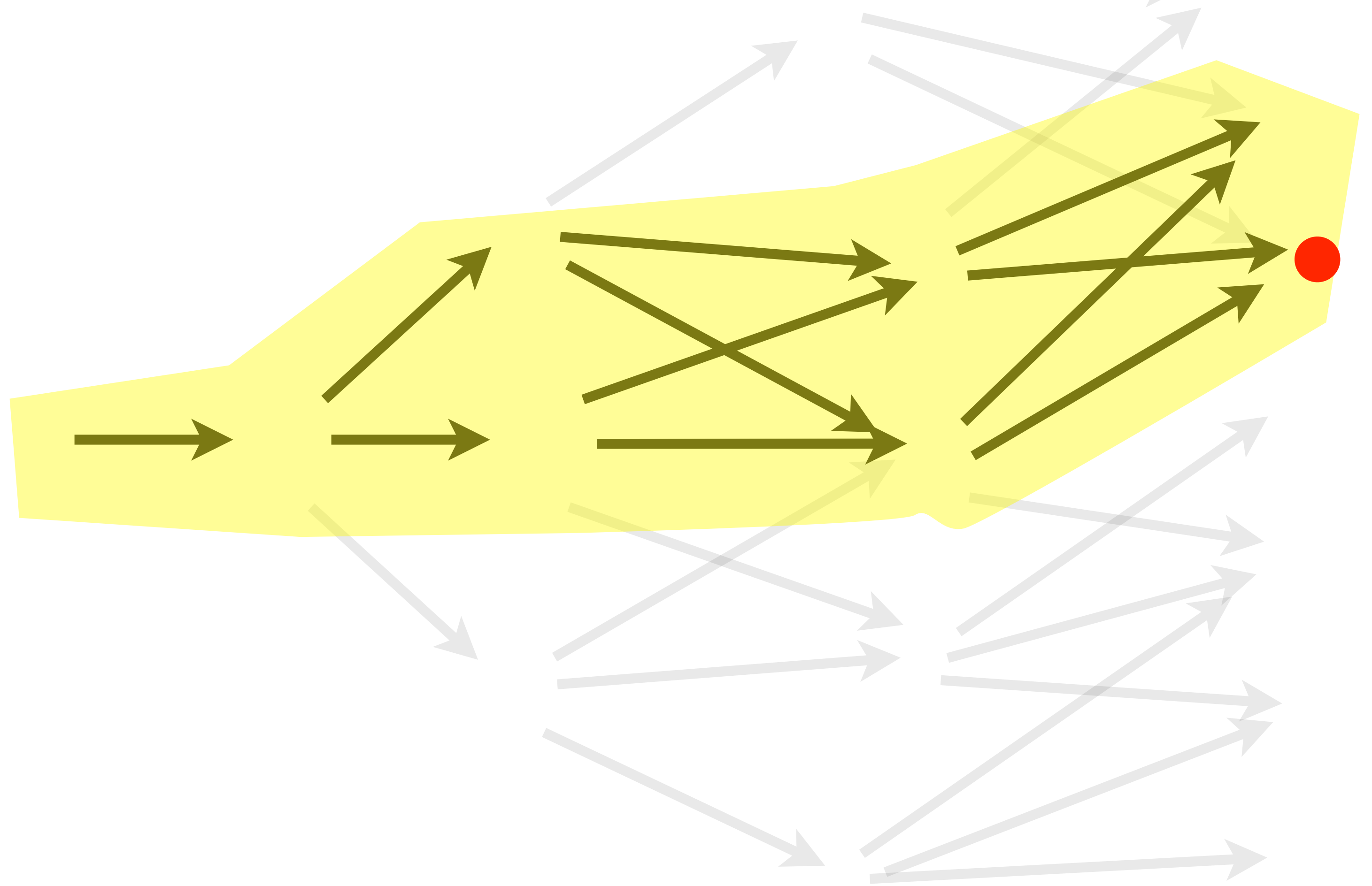
How to Fold RNAs in Linear-Time?

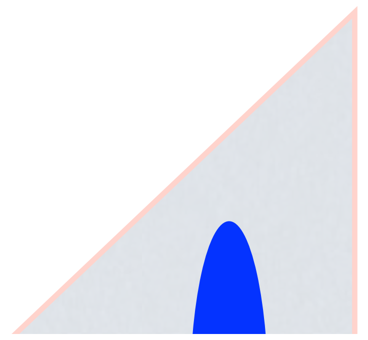
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

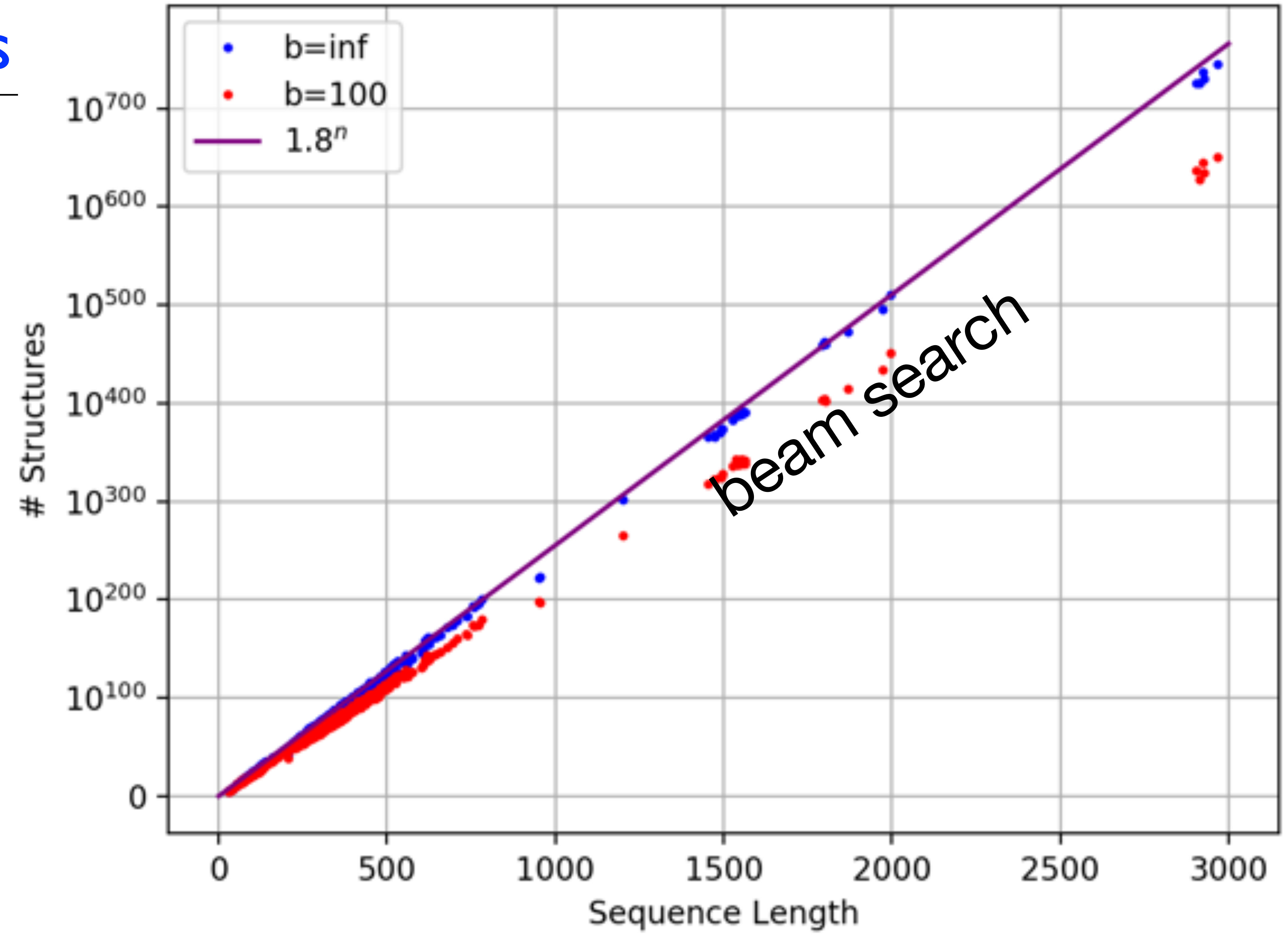
(((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

- idea 2: approximate search: beam pruning
  - keep only top *b* states per step
  - DP+beam: $O(n)$

each DP state corresponds to exponentially many non-DP states

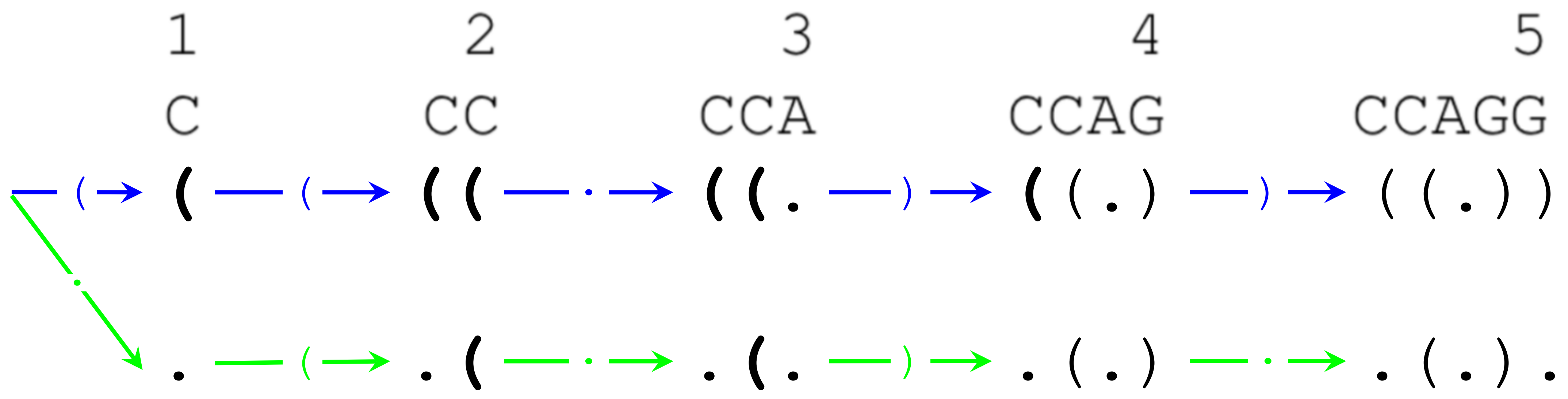






Ambiguity Packing in Biology and Language

- two states are "temporarily equivalent" if their rightmost unpaired brackets are the same



psycholinguistic evidence
(eye-tracking experiments):

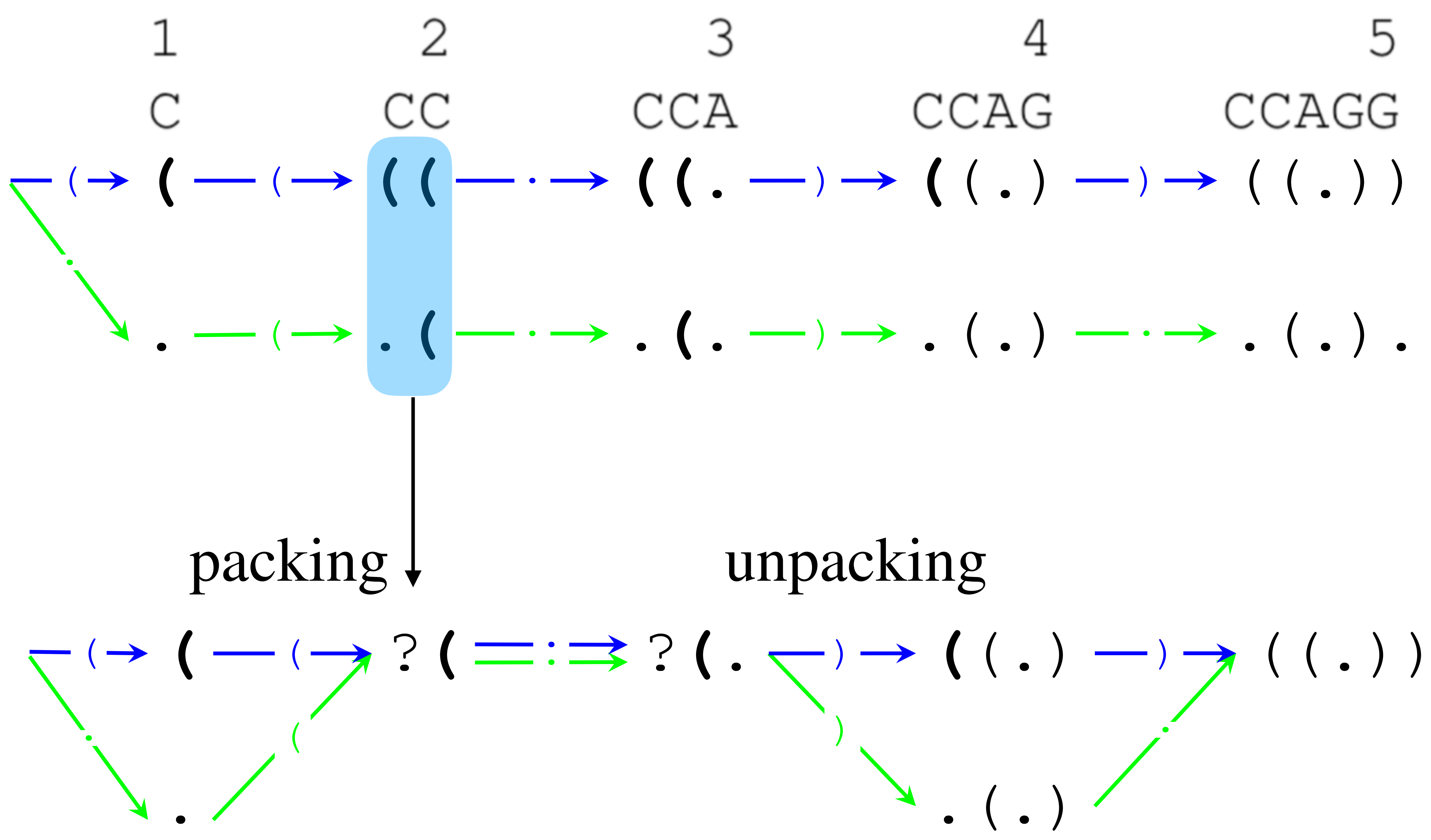
delayed disambiguation

John and Mary had 2 papers
John and Mary had 2 papers

Frazier and Rayner (1990), Frazier (1999)

Ambiguity Packing in Biology and Language

- two states are "temporarily equivalent" if their rightmost unpaired brackets are the same



psycholinguistic evidence
(eye-tracking experiments):

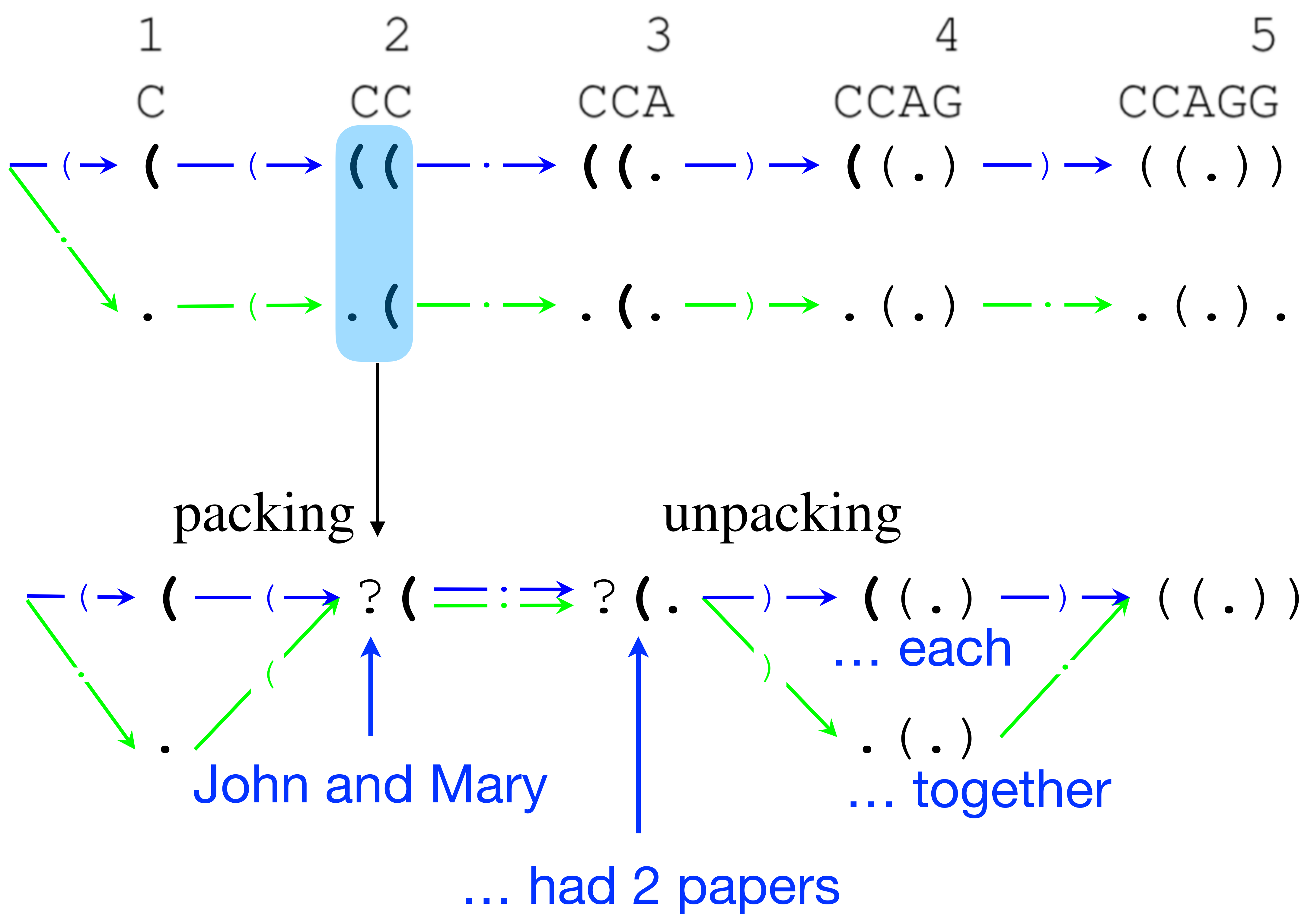
delayed disambiguation

John and Mary had 2 papers
John and Mary had 2 papers

Frazier and Rayner (1990), Frazier (1999)

Ambiguity Packing in Biology and Language

- two states are "temporarily equivalent" if their rightmost unpaired brackets are the same



psycholinguistic evidence
(eye-tracking experiments):

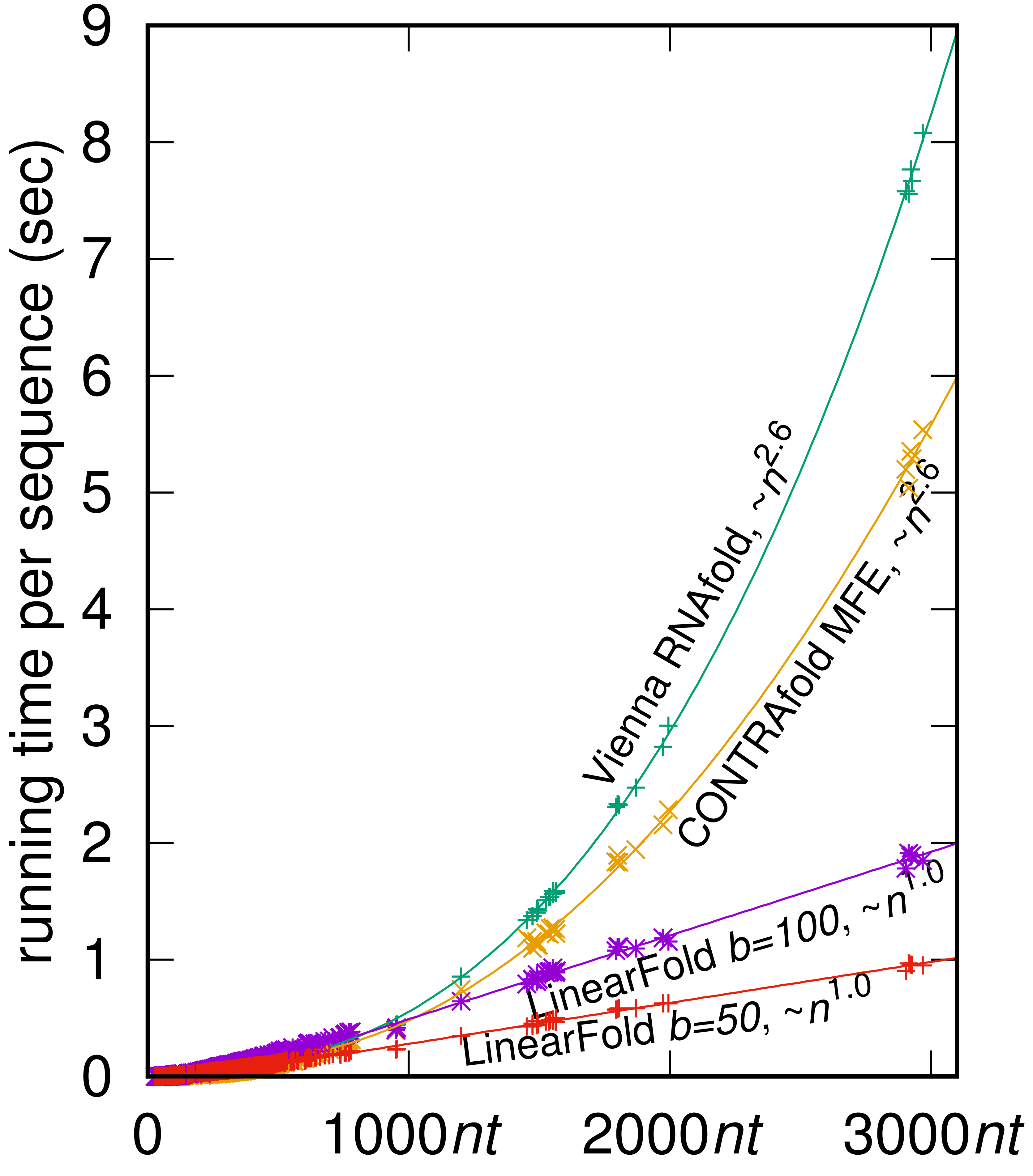
delayed disambiguation

John and Mary had 2 papers each
John and Mary had 2 papers together

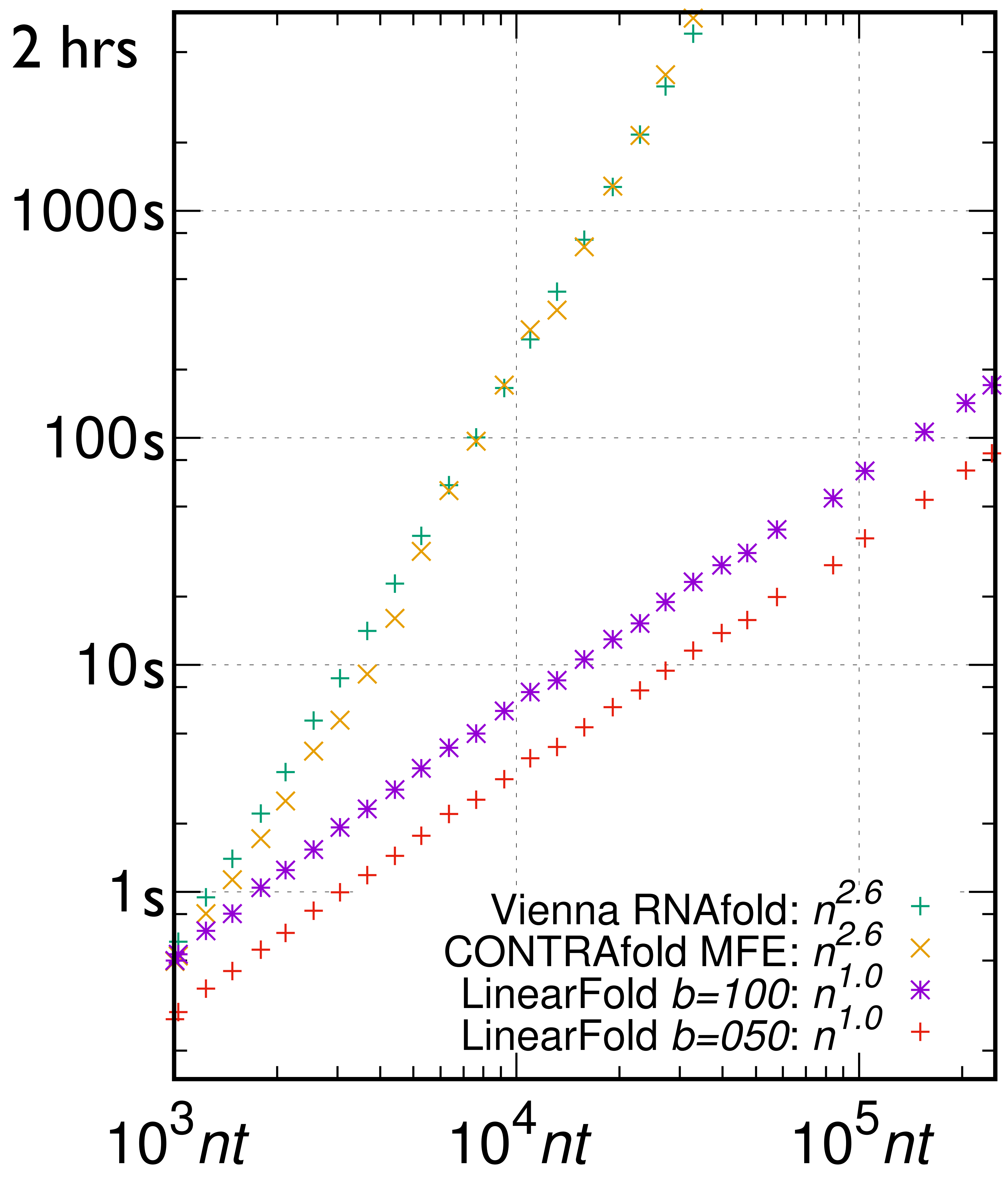
Frazier and Rayner (1990), Frazier (1999)

Our Linear-Time Prediction is Much Faster…

10,000*nt* (~HIV)
4min → 7s

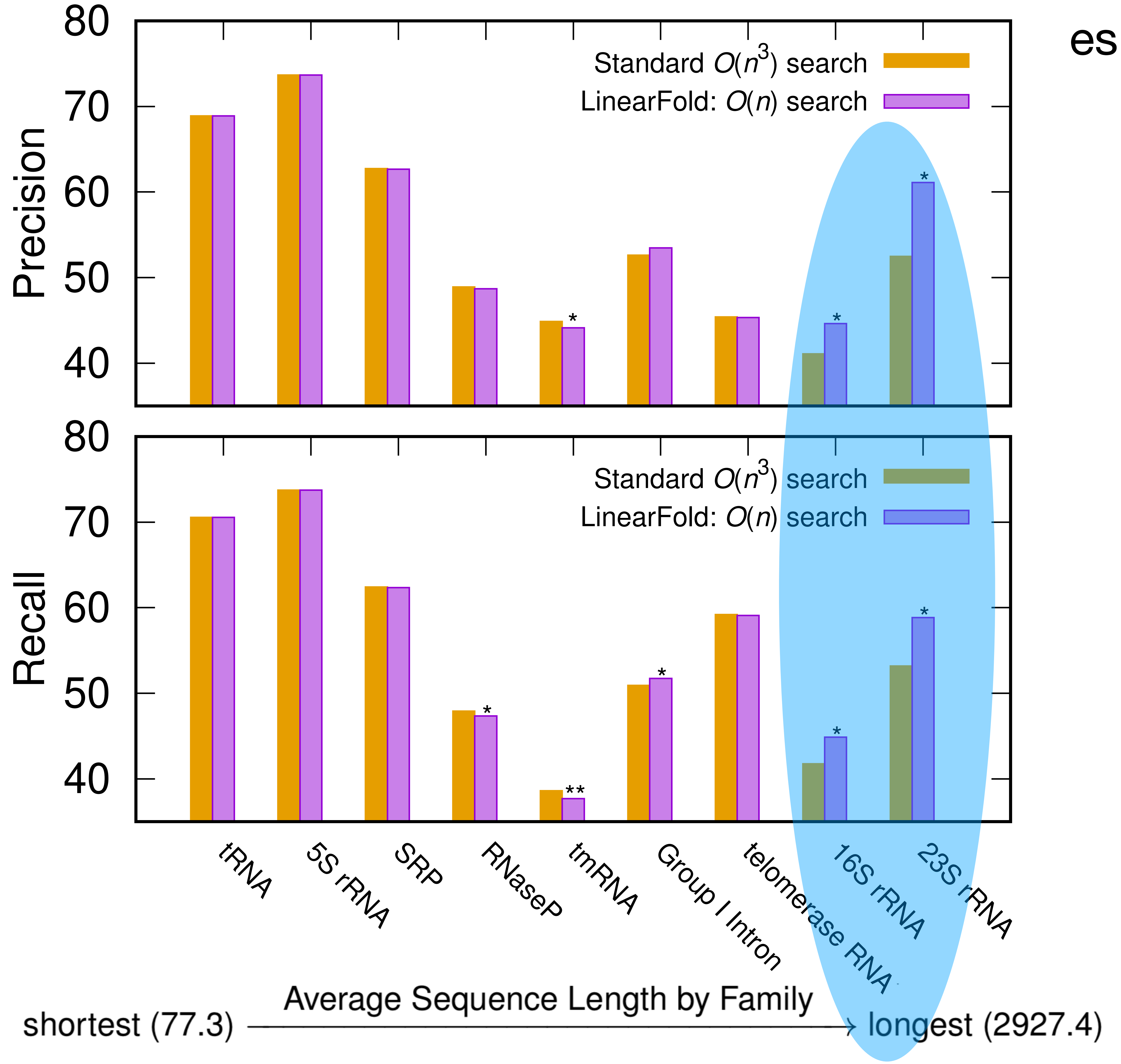


244,296*nt* (longest in RNAcentral)
~200hrs → 120s



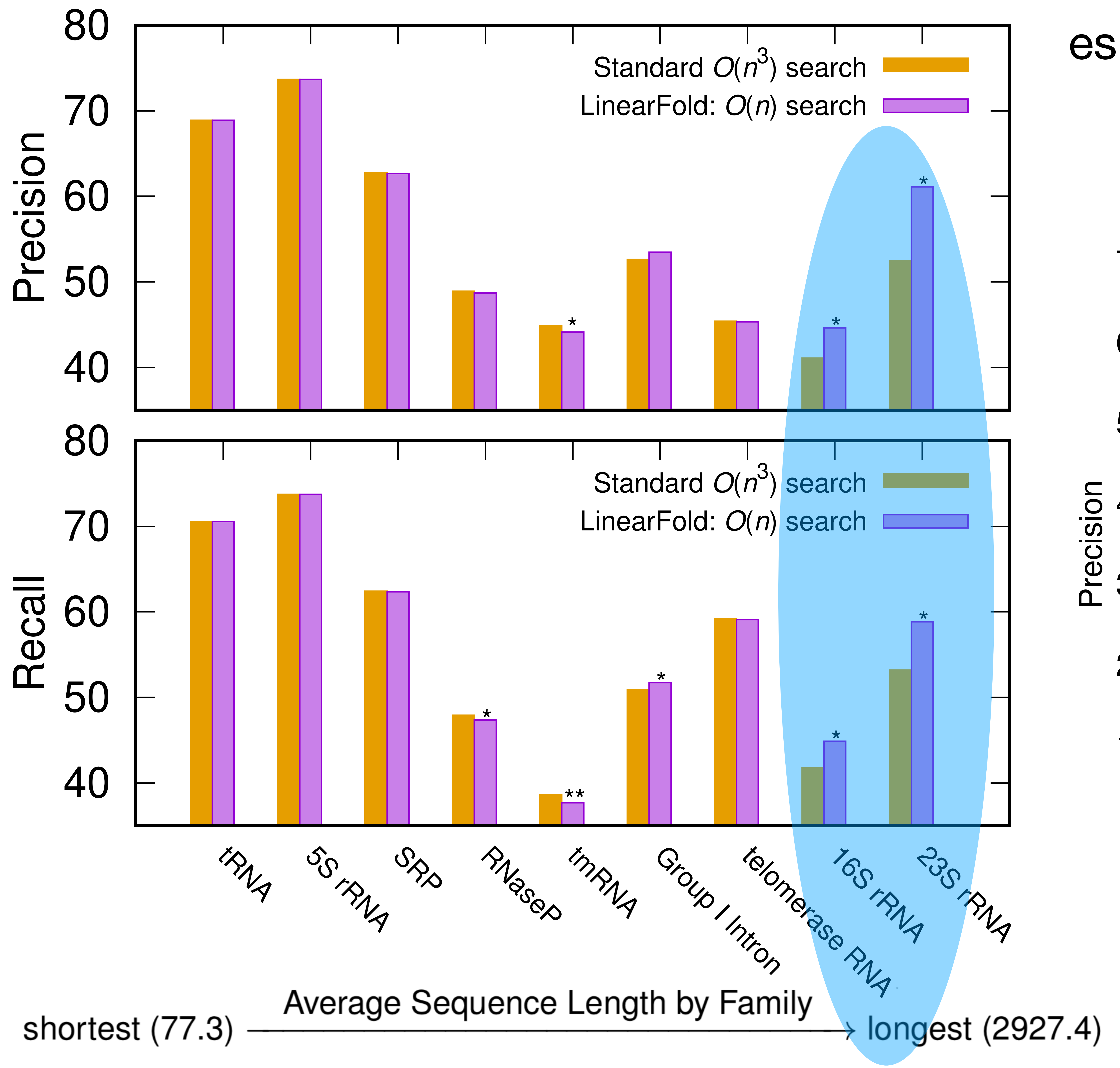
with even slightly better prediction accuracy!!

… and Also More Accurate!

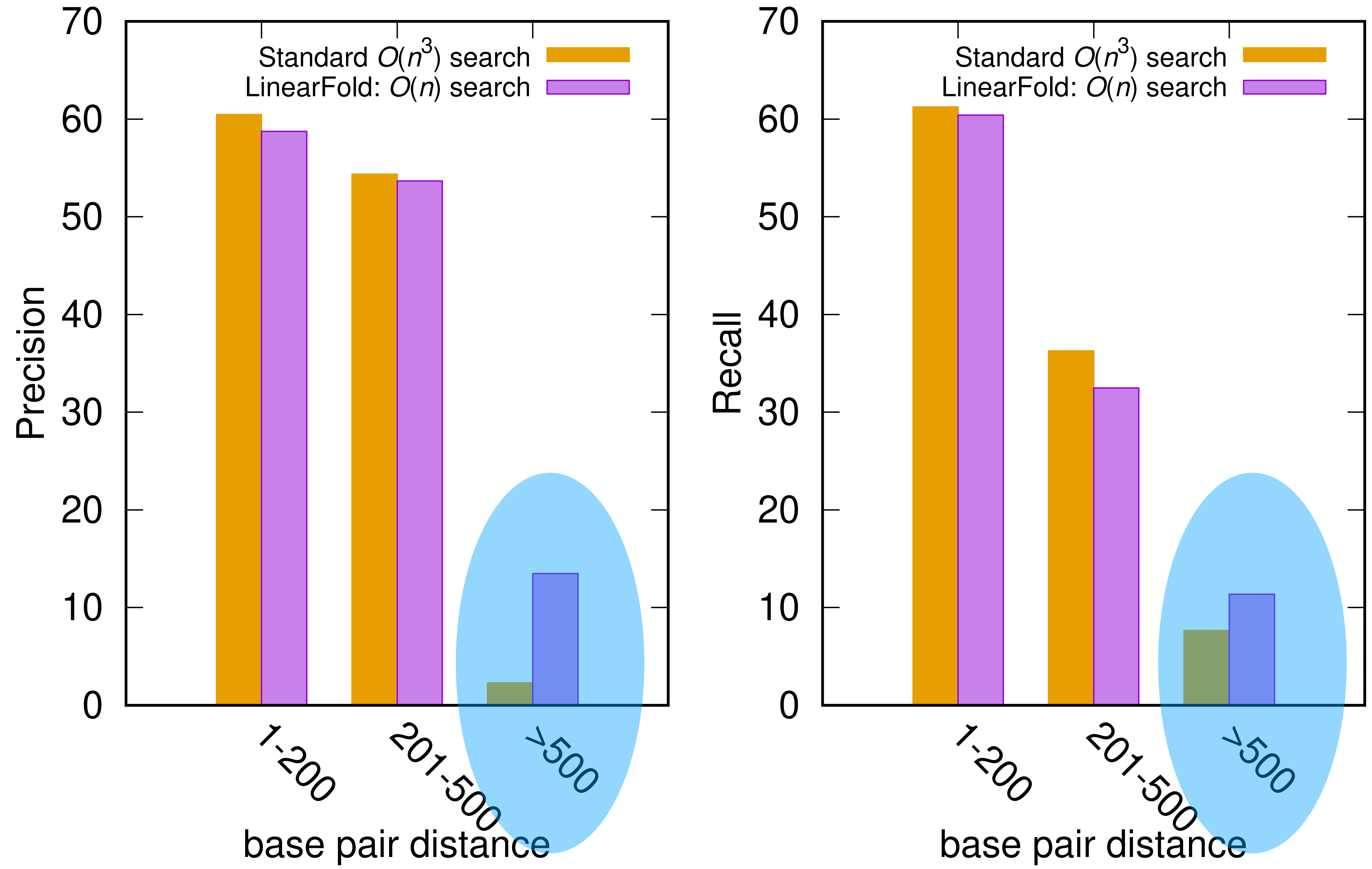


esp. on longer sequences and long-range base pairs

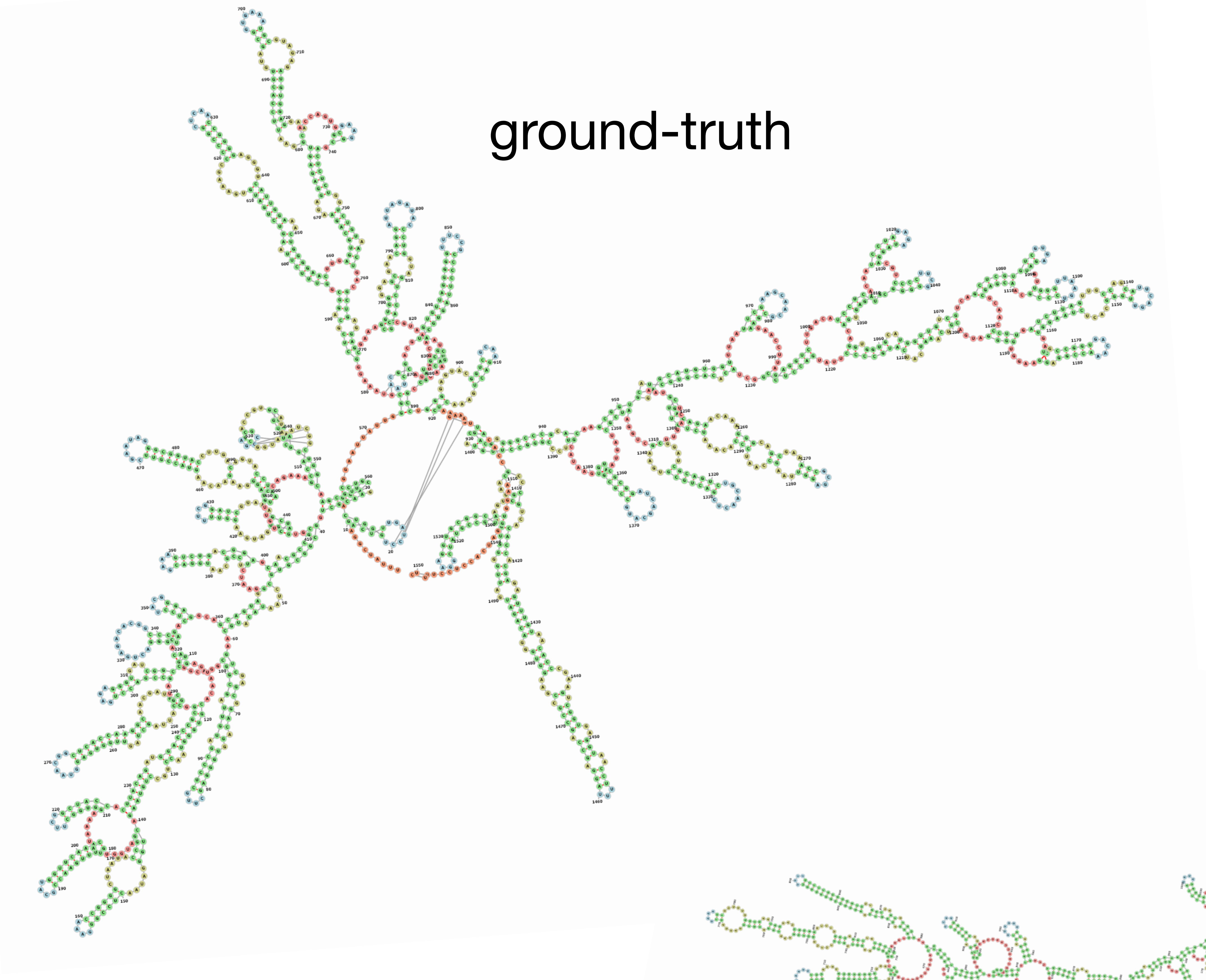
… and Also More Accurate!



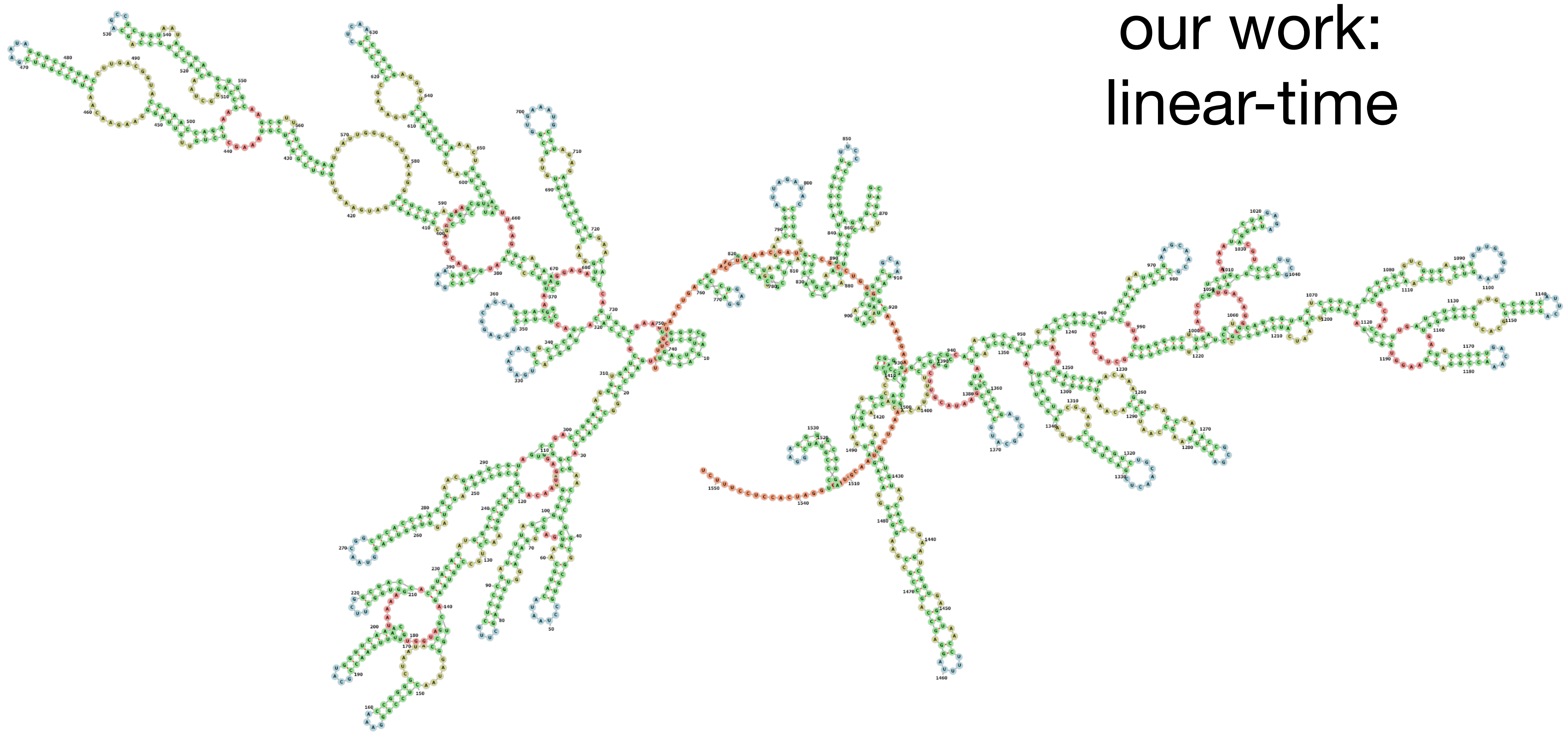
esp. on longer sequences and long-range base pairs



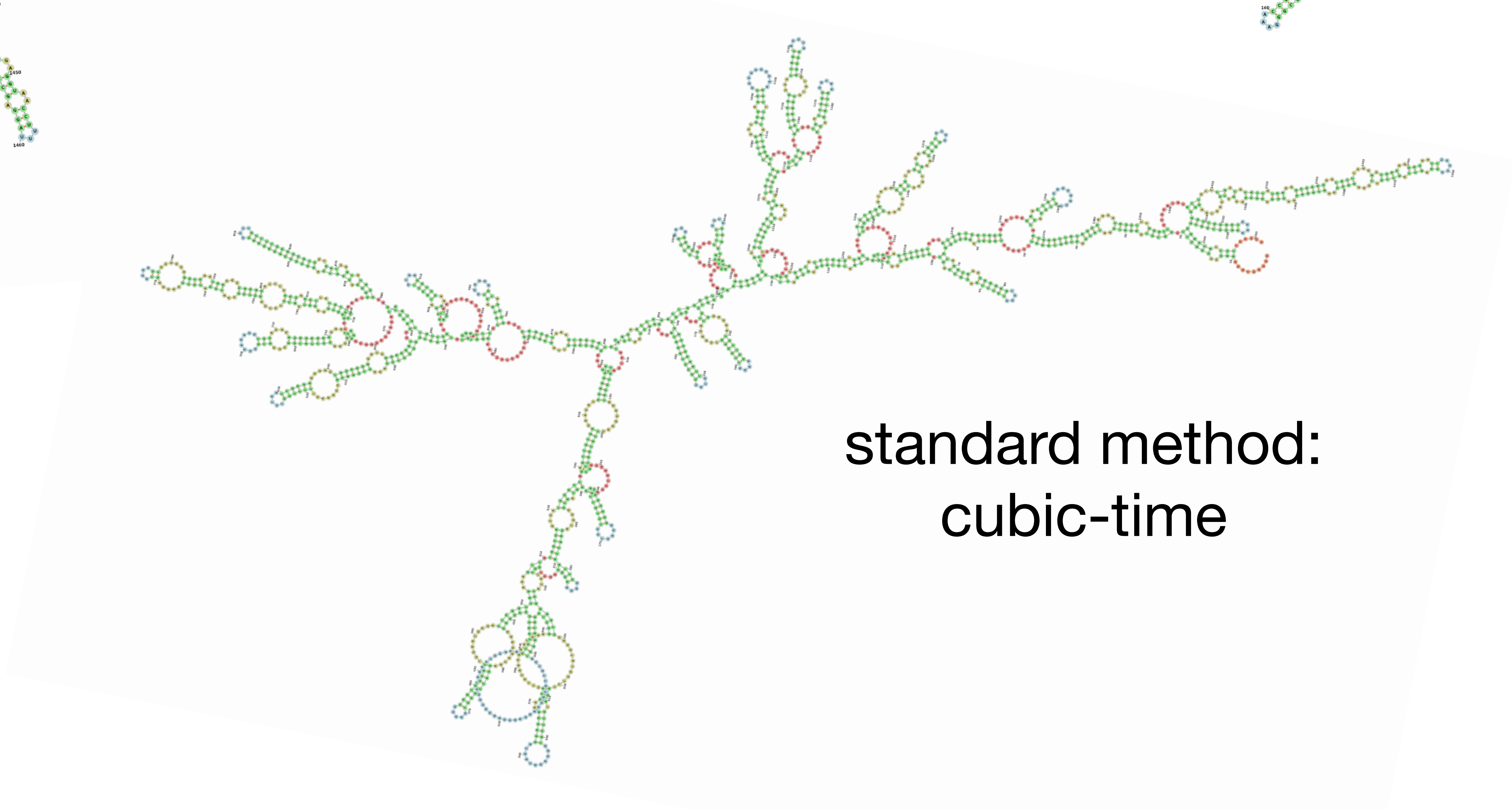
Example: *B. Subtilis* 16S rRNA (length: 1,552*nt*)



ground-truth

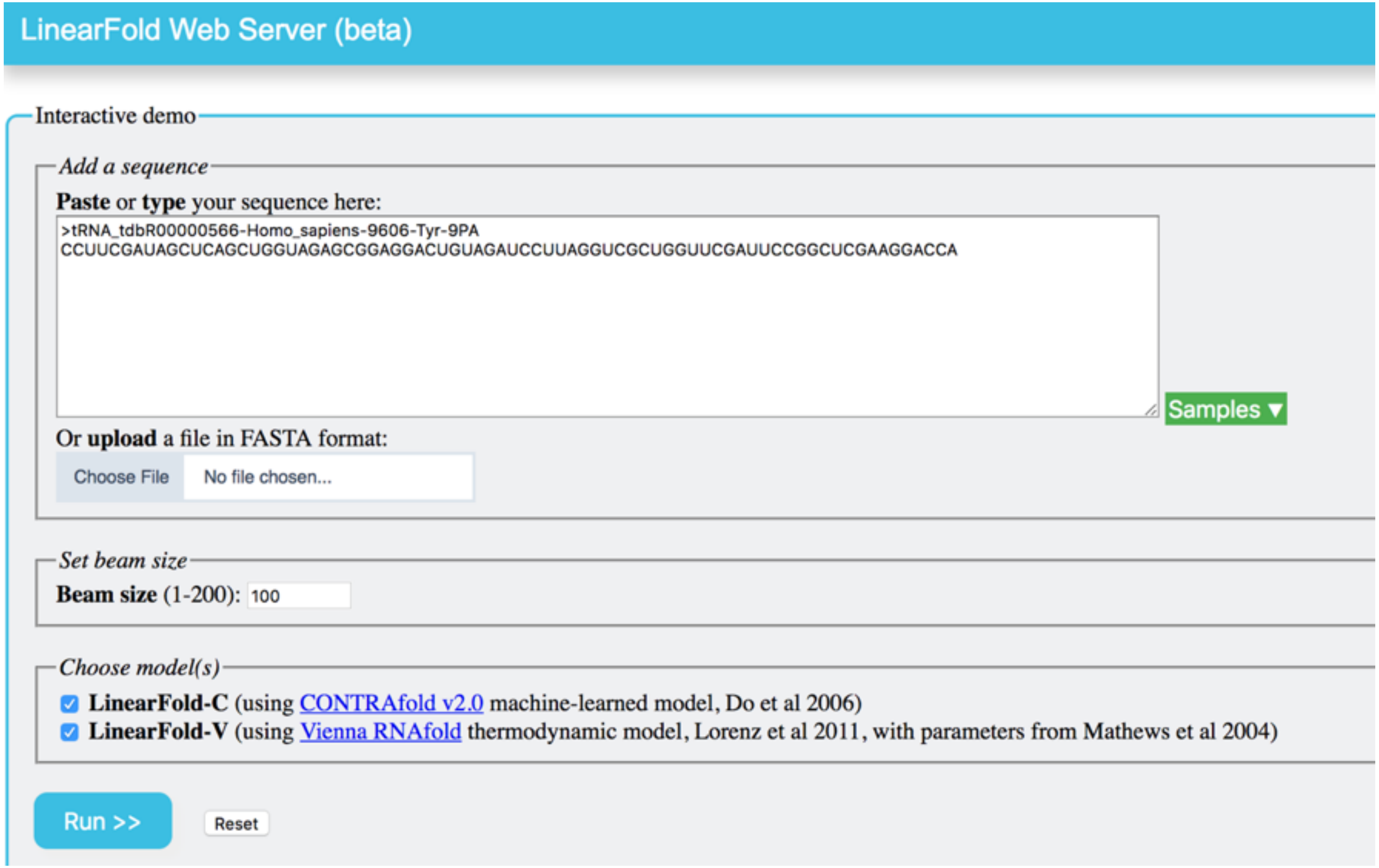


our work:
linear-time



standard method:
cubic-time

World's Fastest RNA Structure Prediction Server



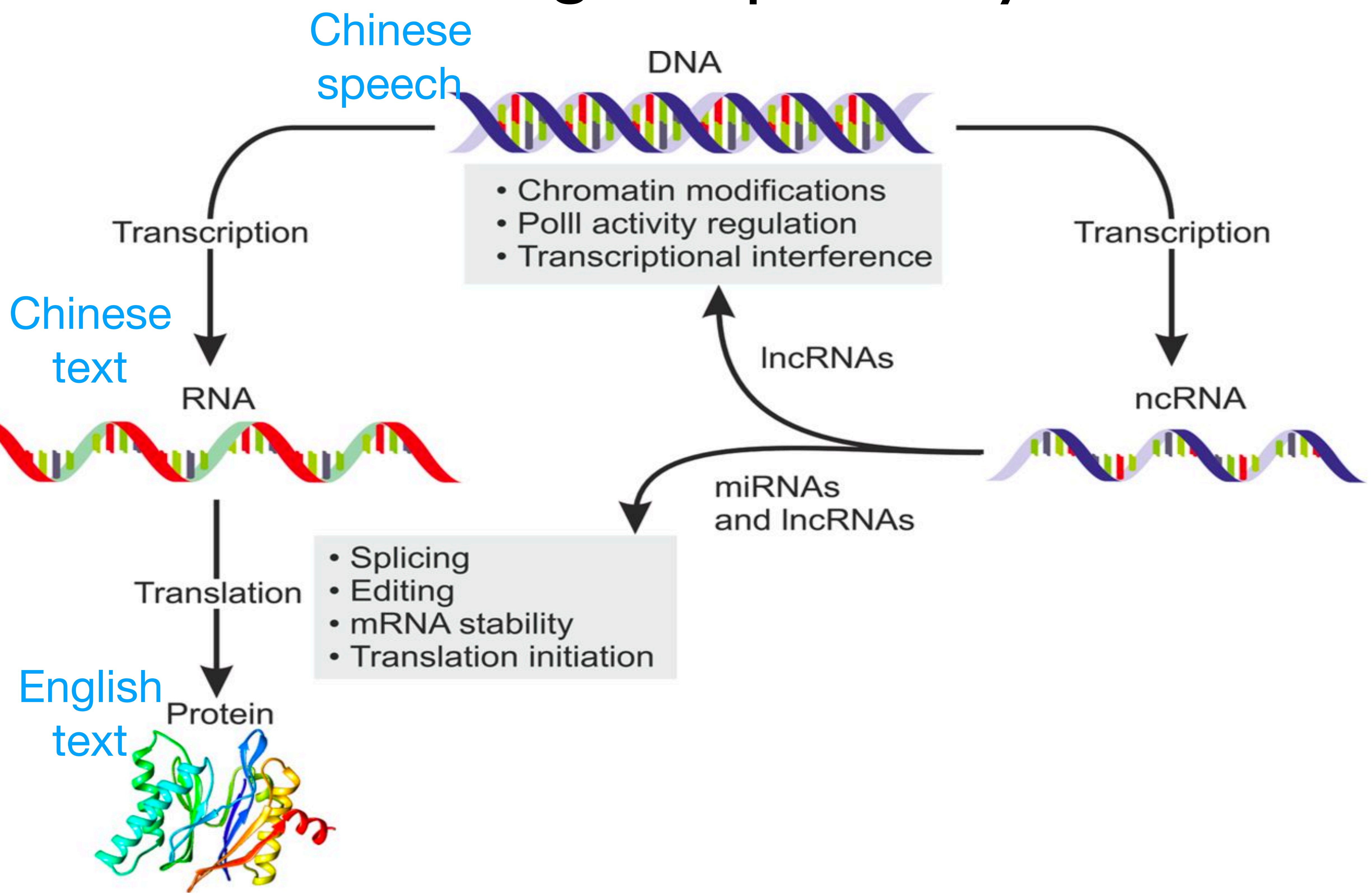
http://linearfold.eecs.oregonstate.edu:8080/

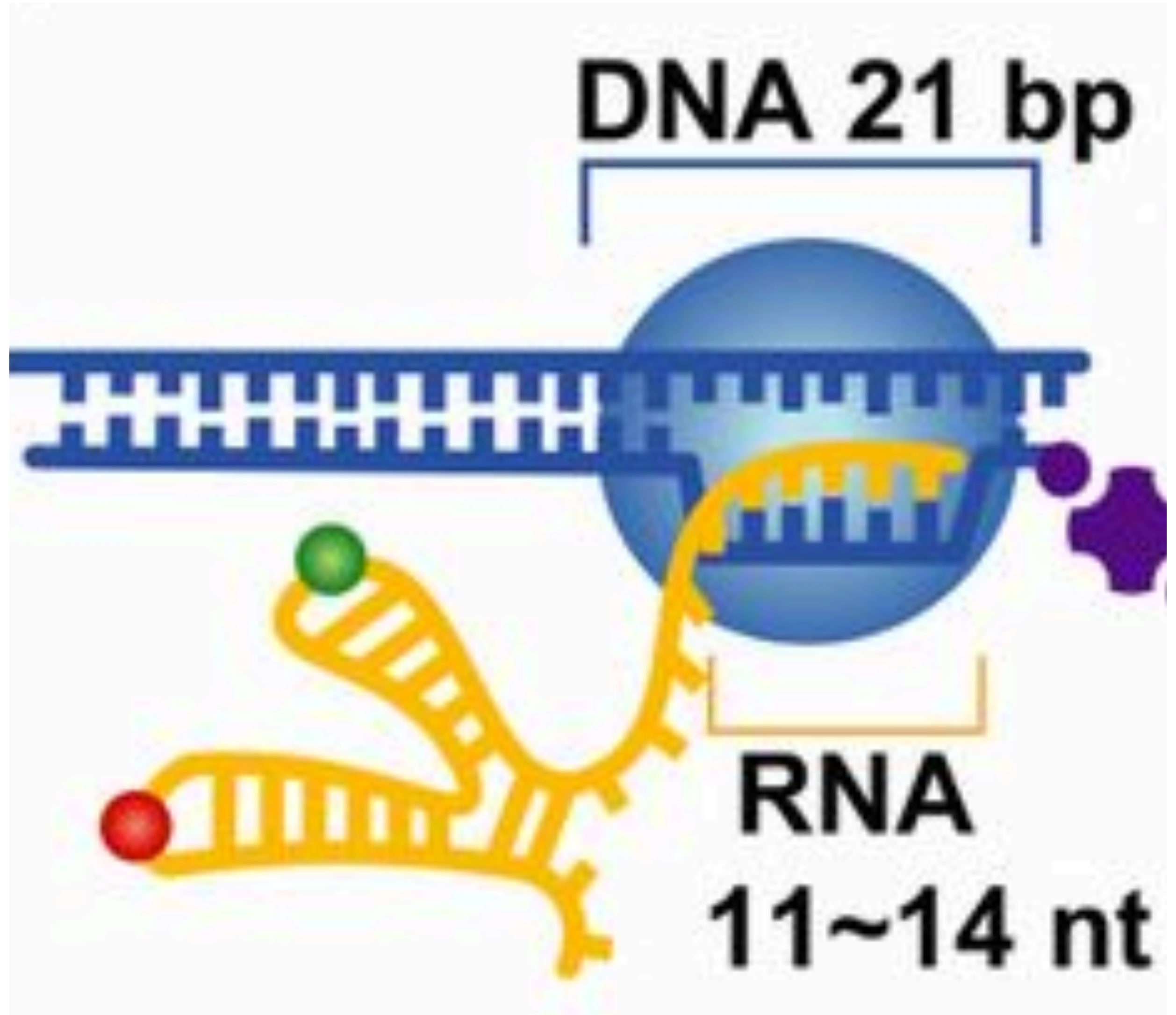
Incremental Parsing <=> Incremental Folding

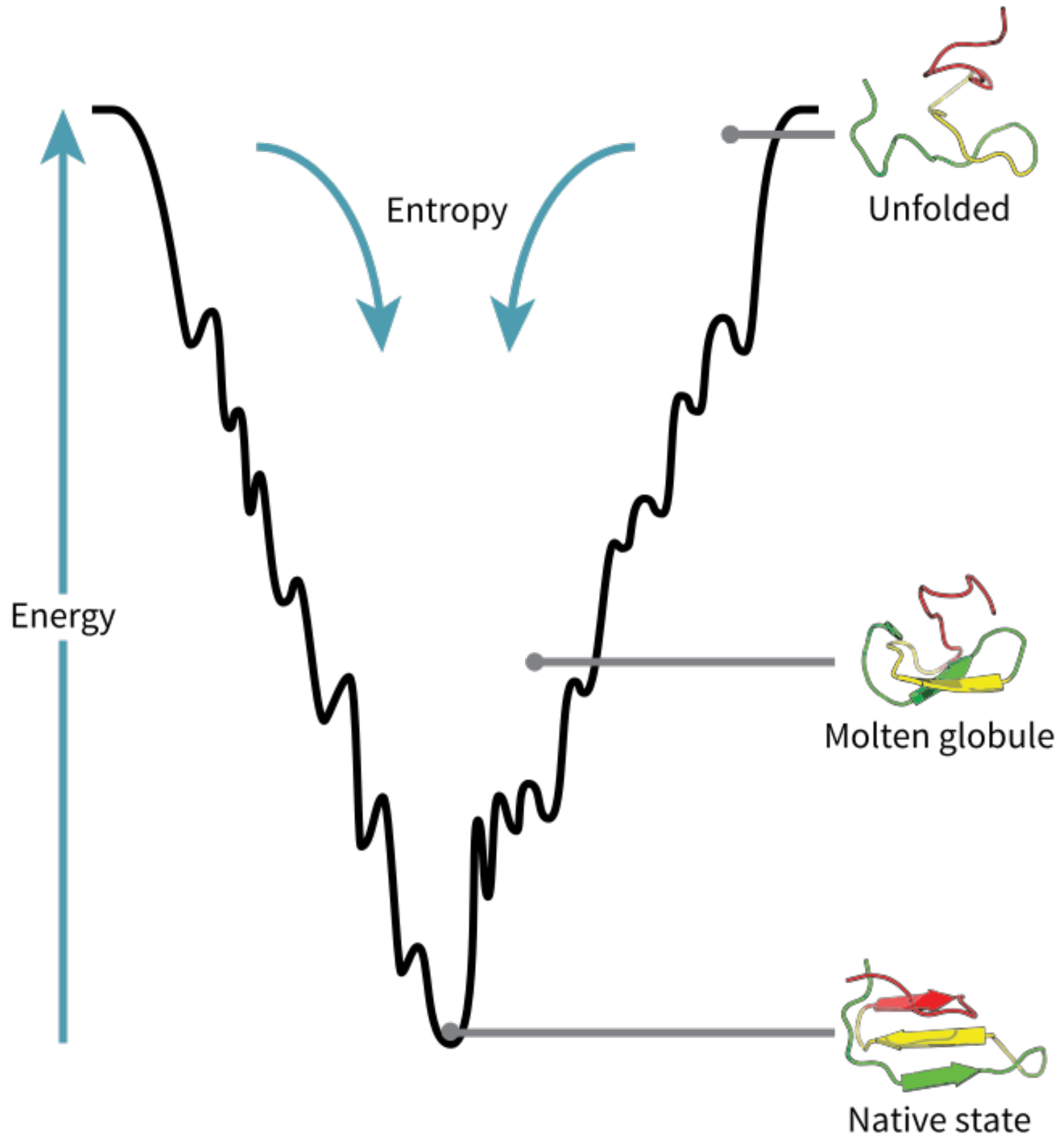
- humans process sentences incrementally
- human language sentences *evolve* to be incrementally parsable

- RNAs & proteins fold while being assembled
- RNA & protein sequences evolve to be incrementally foldable

- these might explain why linear-time search performs better than exact search

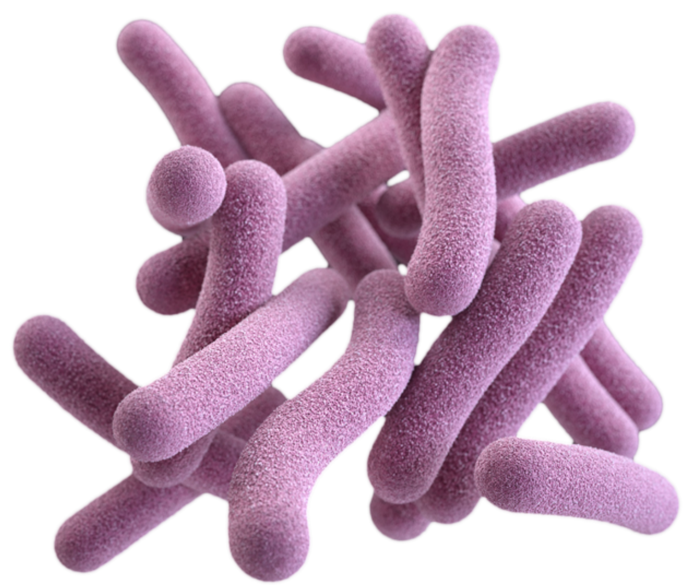


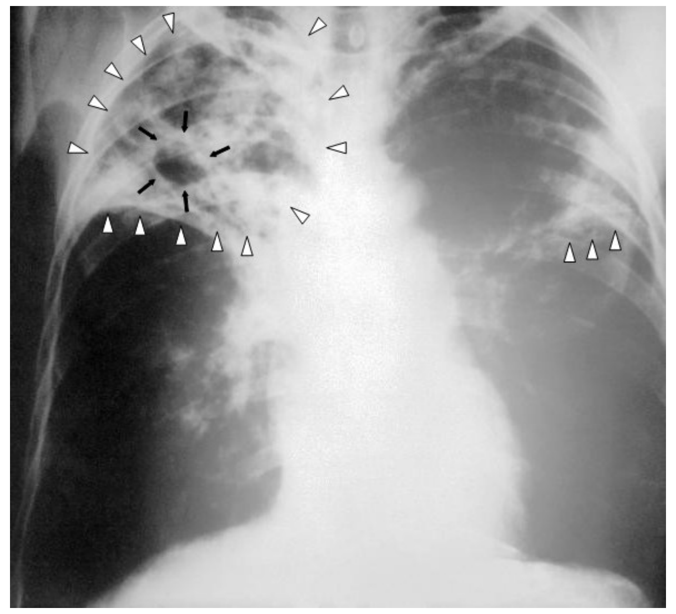




Fast Structure Prediction Enables RNA Design

detecting active TB using RNA design
which needs our fast RNA folding





Professor **Rhiju Das**
Stanford Medical School

EteRNA game
(RNA design)

RNA sequence

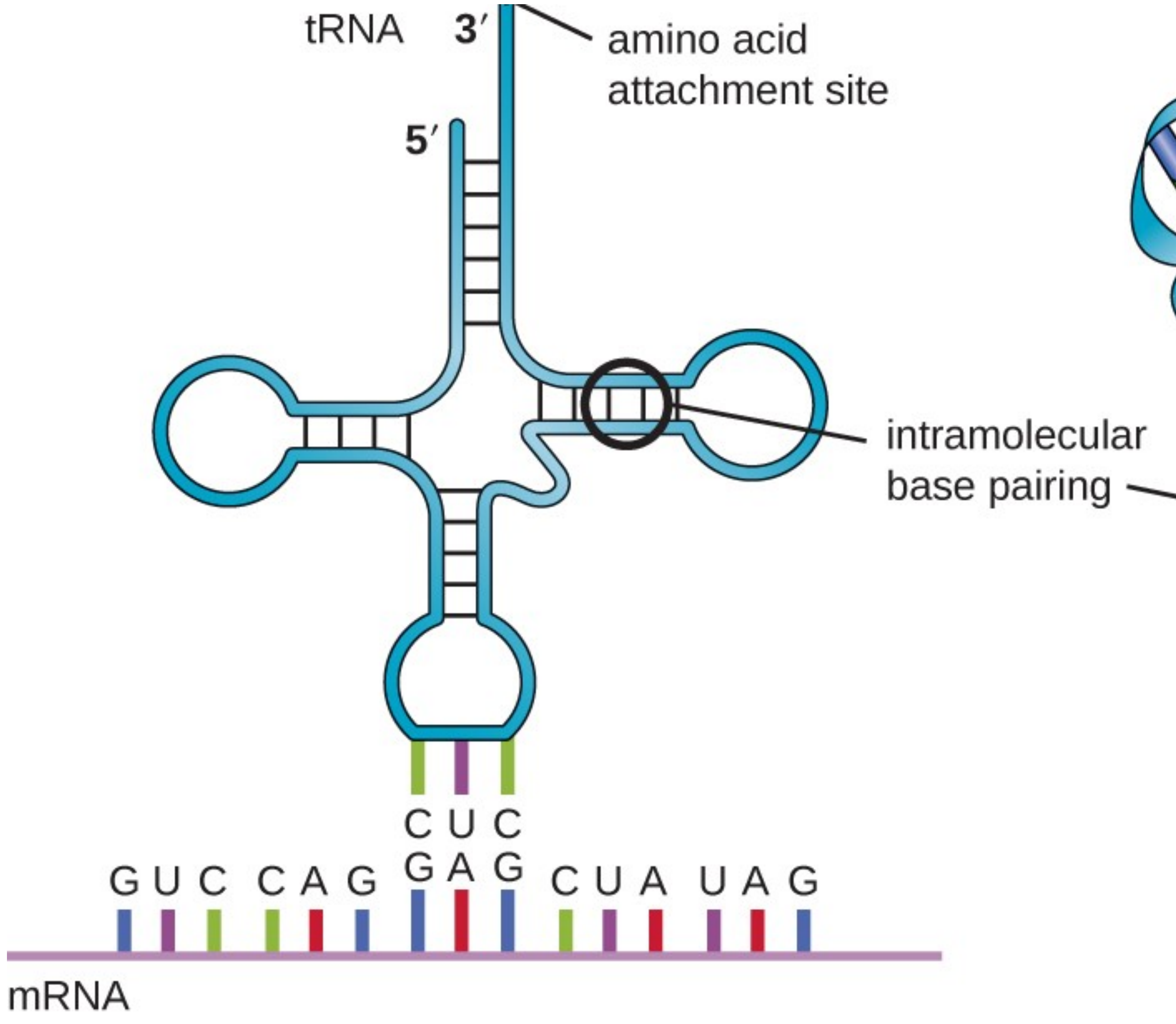
GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

structure prediction
("folding")

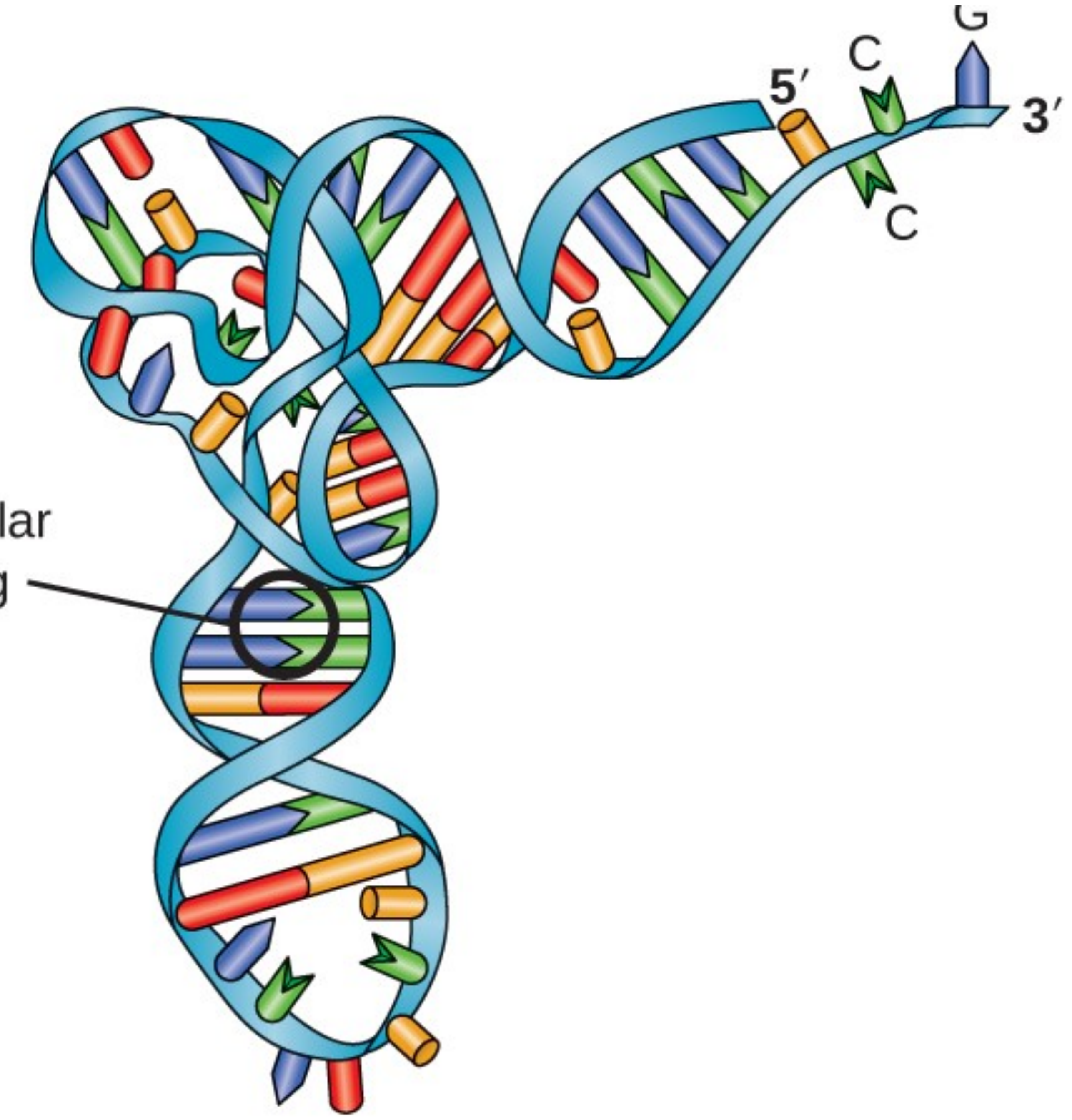
design

RNA secondary structure

RNA 3D structure



tRNA 2-dimensional structure



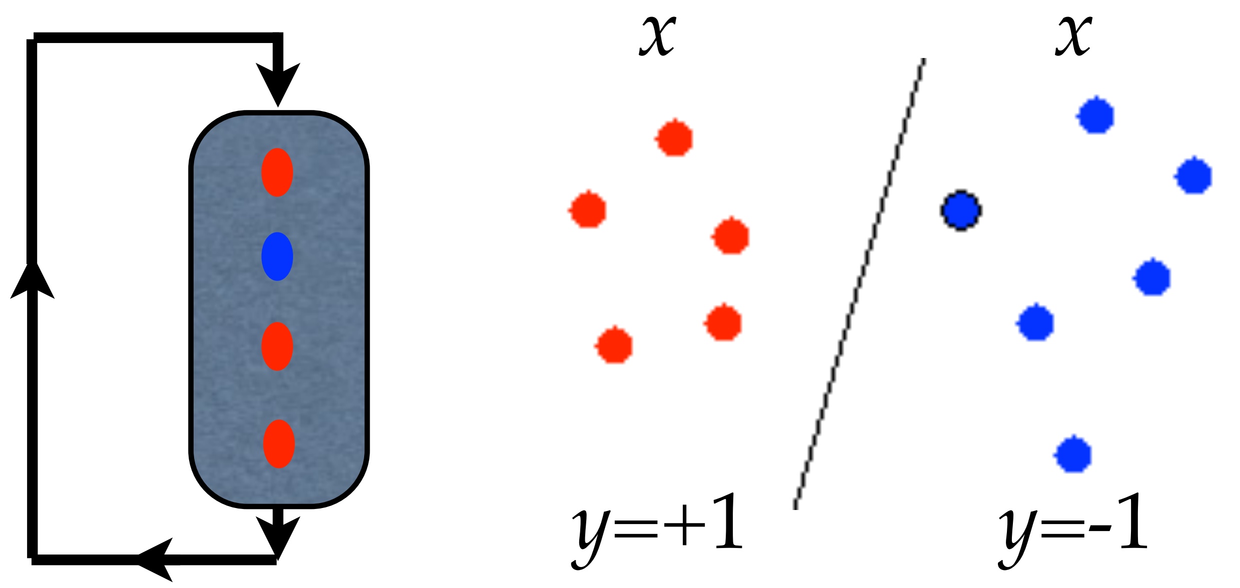
tRNA 3-dimensional structure

Other Work, Recap, and Vision

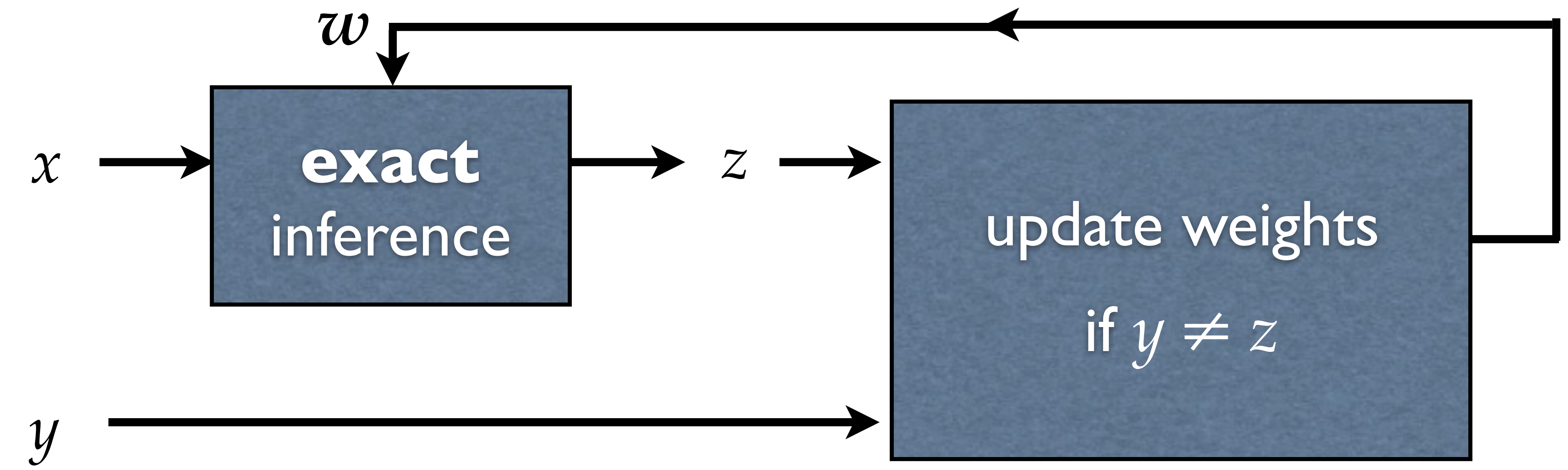
Other Work, Recap, and Vision

Other Work: Structured Prediction: Linear-Time Learning

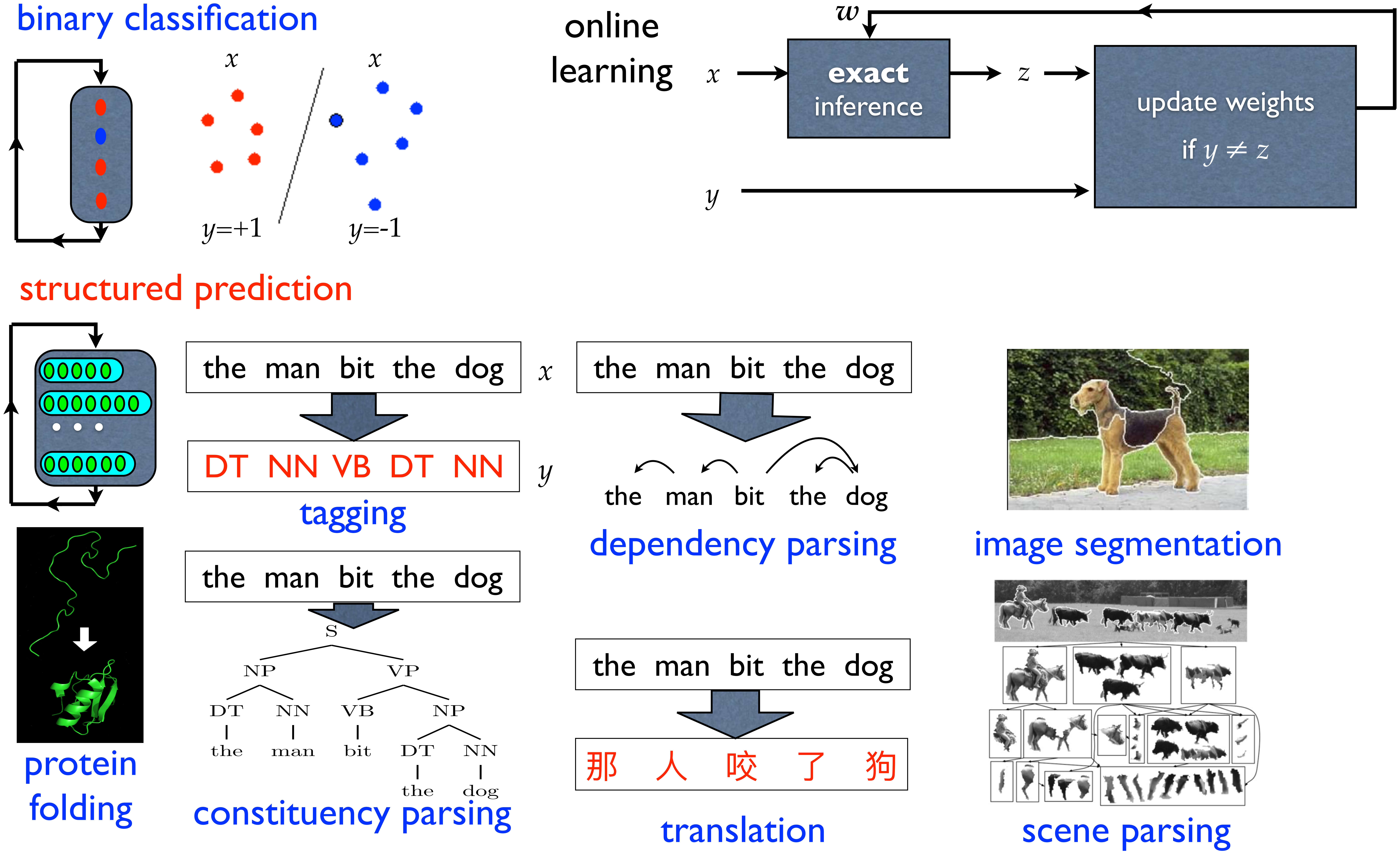
binary classification



online learning

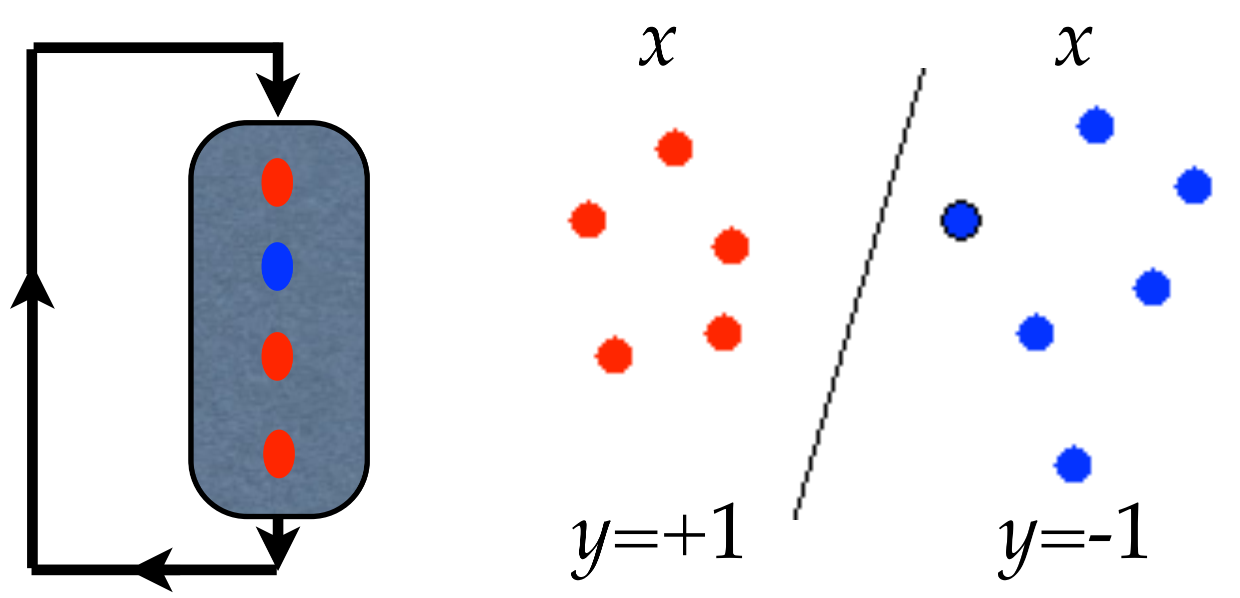


Other Work: Structured Prediction: Linear-Time Learning

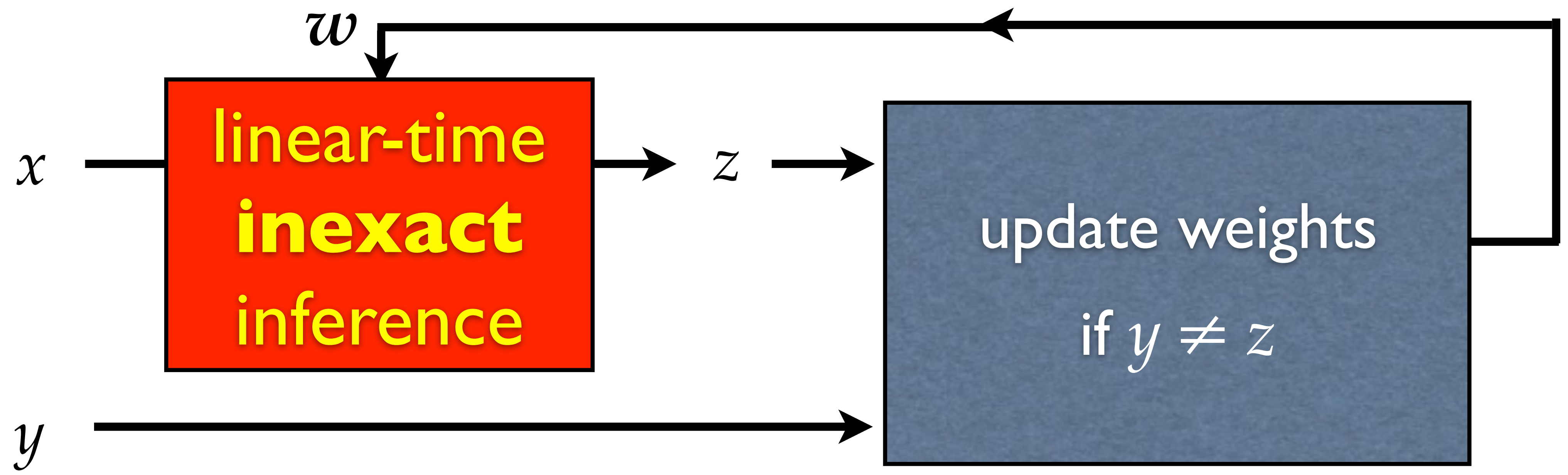


Other Work: Structured Prediction: Linear-Time Learning

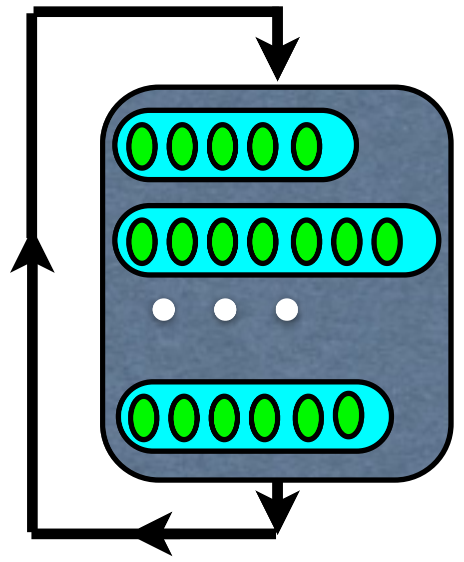
binary classification

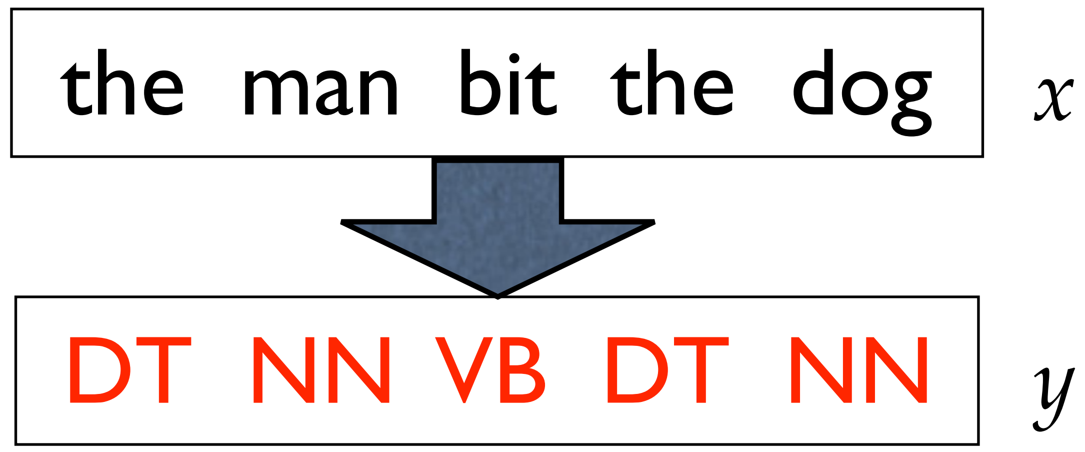


online learning

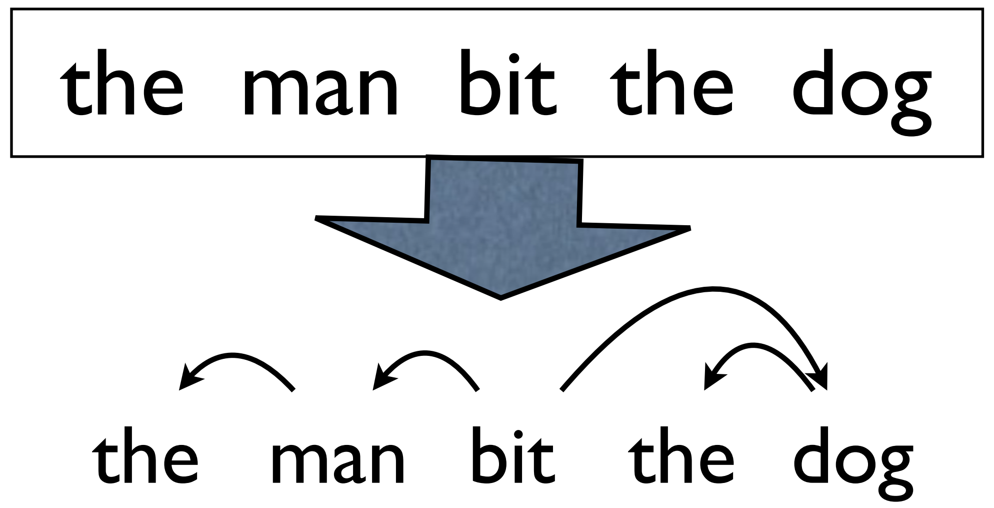


structured prediction



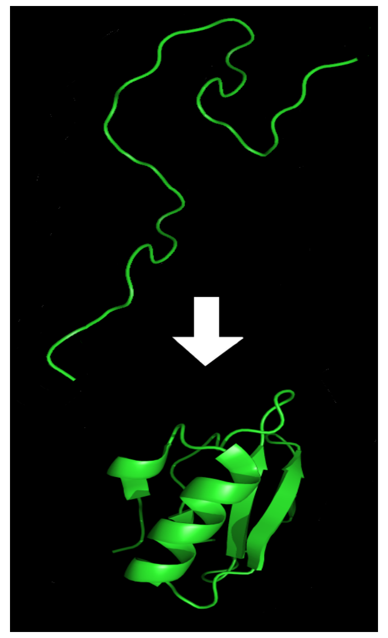


tagging

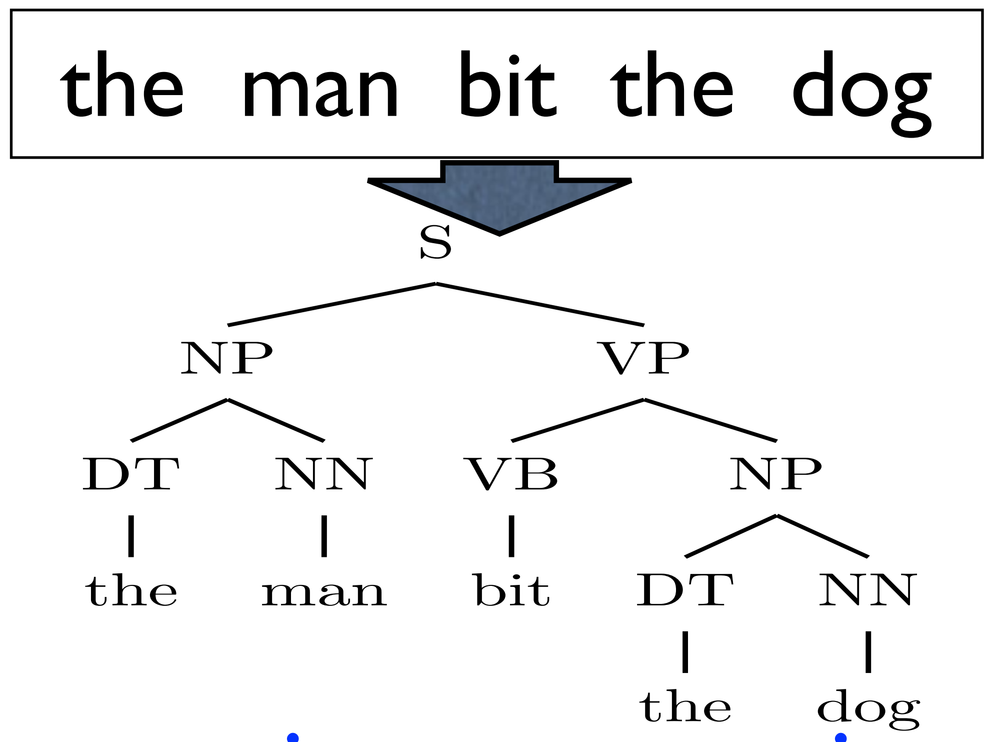


dependency parsing

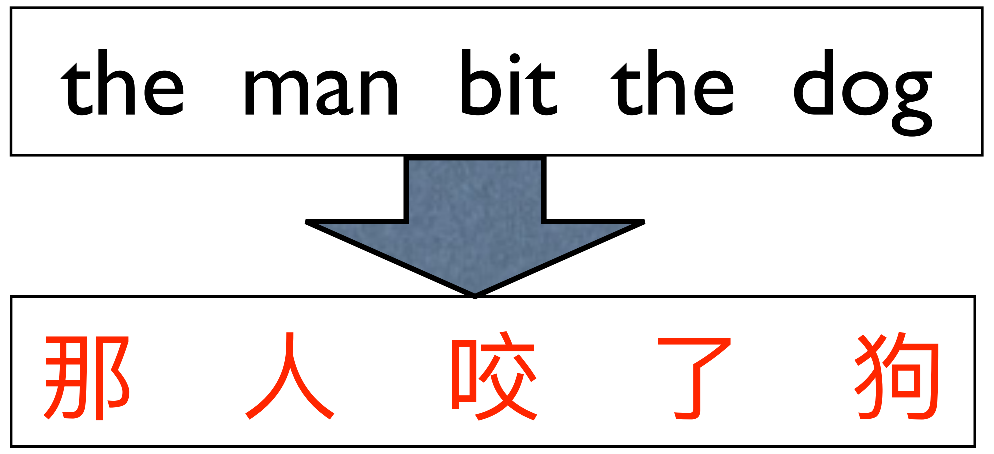
image segmentation



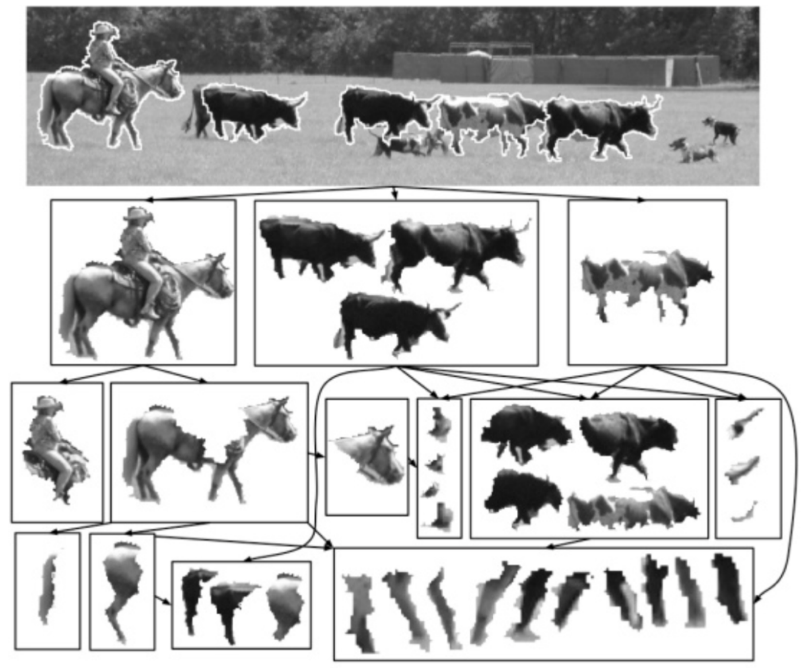
protein folding



constituency parsing



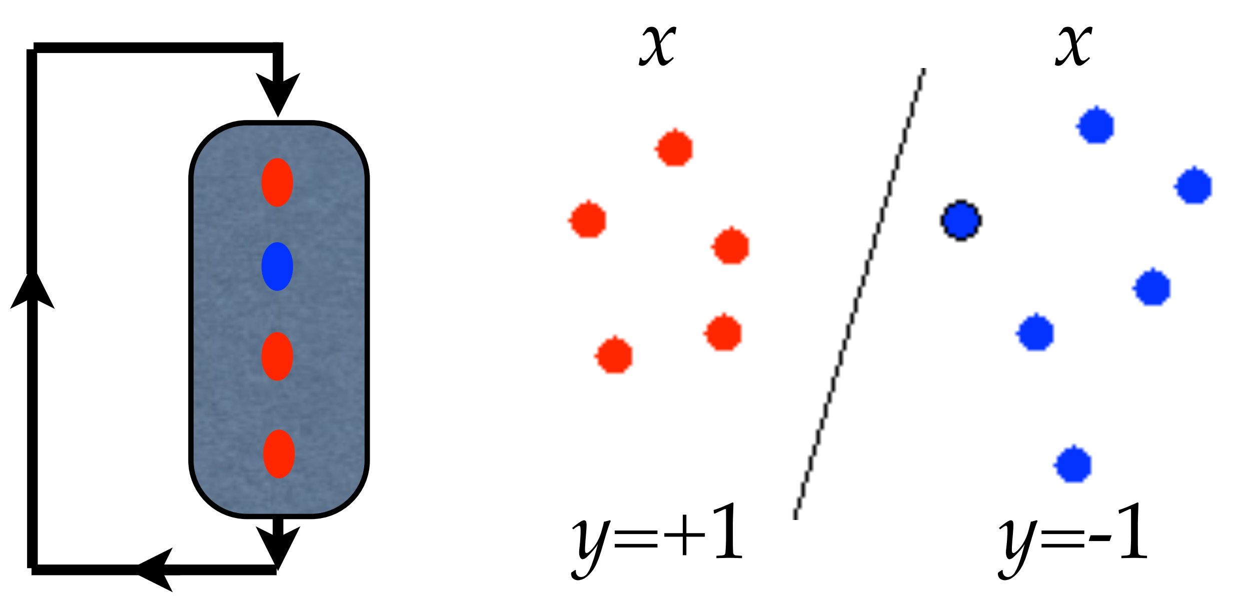
translation



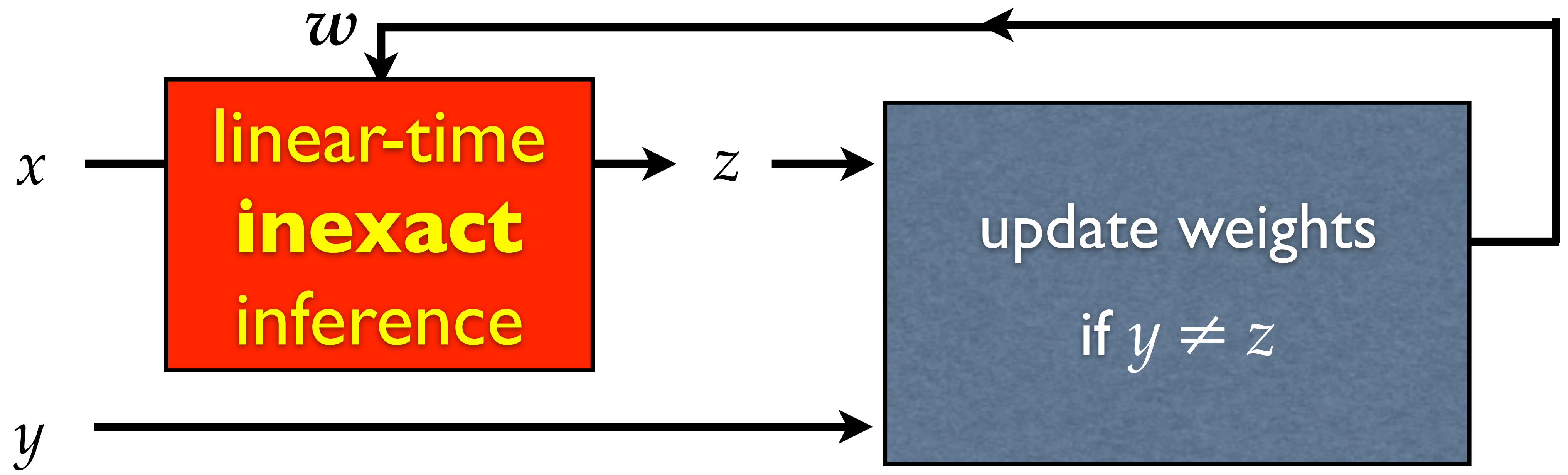
scene parsing

Other Work: Structured Prediction: Linear-Time Learning

binary classification

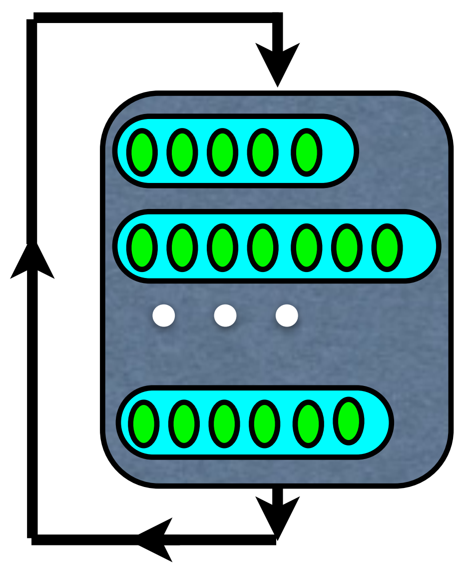


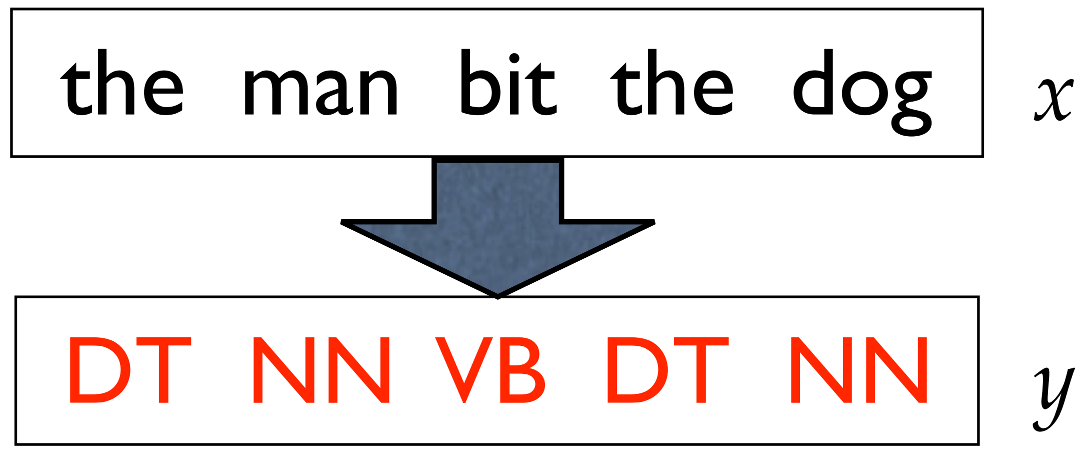
online learning



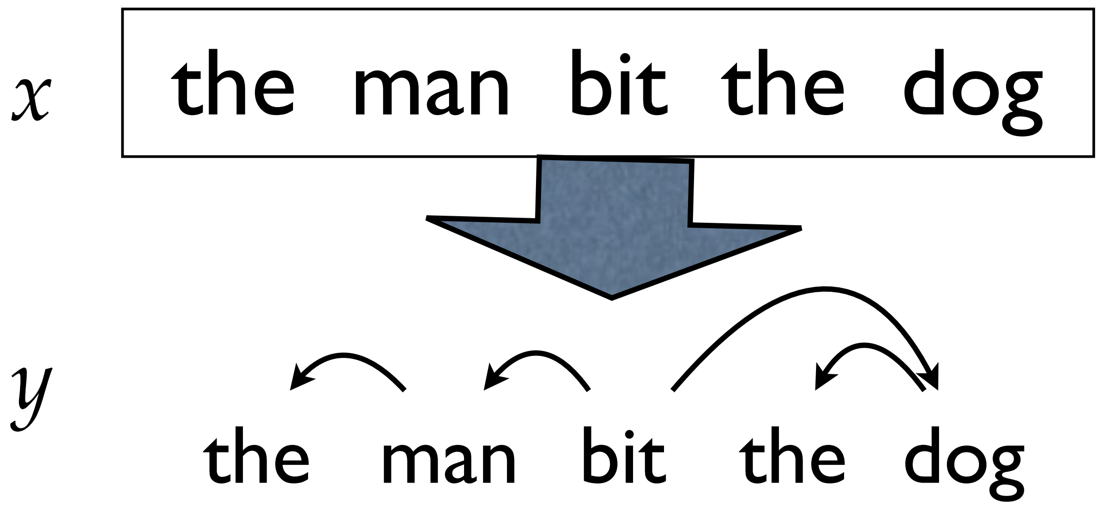
new convergence theorems:
(modified) online learning still converges with inexact search

structured prediction



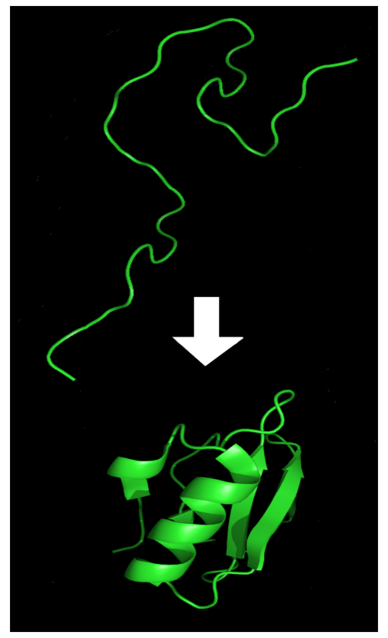


tagging

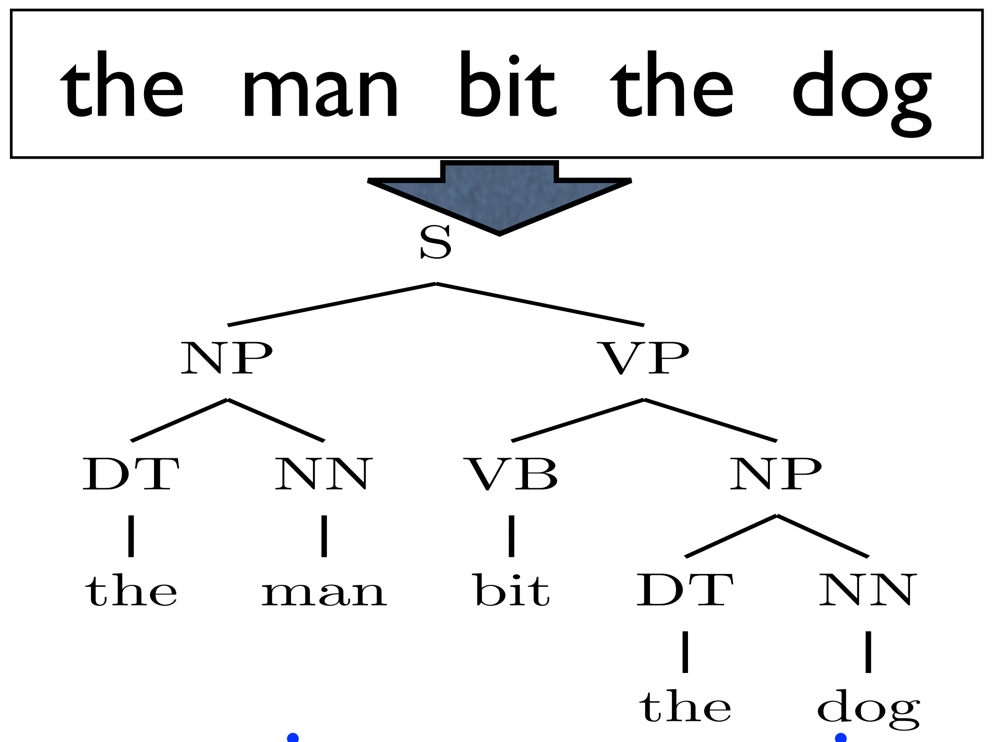


dependency parsing

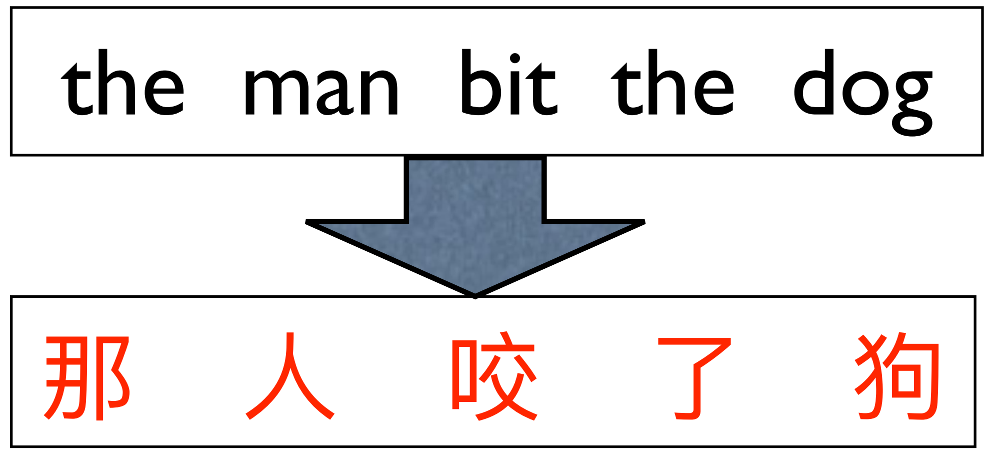
image segmentation



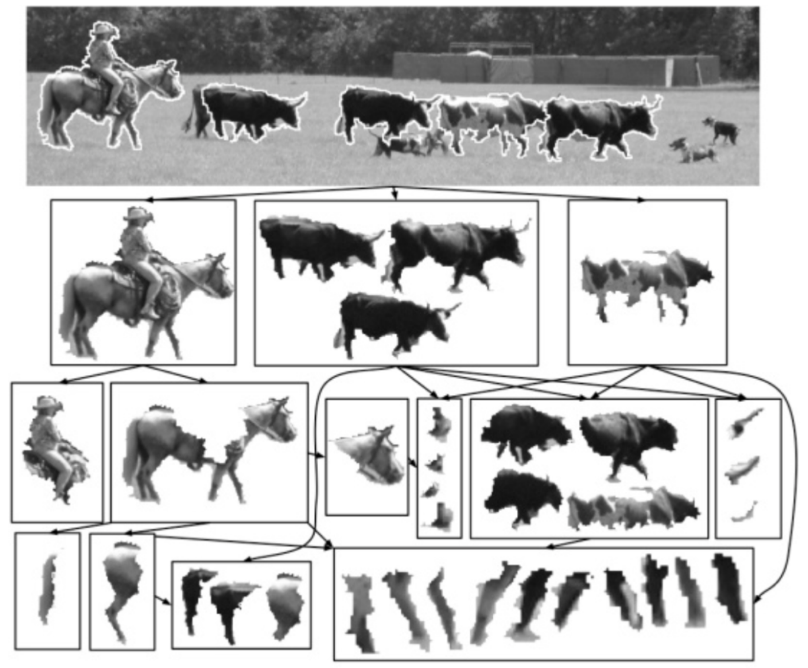
protein folding



constituency parsing



translation



scene parsing

Other Work: Incremental Semantic Parsing

- parse NL (e.g., English) into a formal meaning representation
  - type-driven parsing (formal semantics) + polymorphism (PL theory)
  - future work: (simultaneously) translating NL into PL (e.g., SQL)

What is the capital of the largest state by area?

step	action	stack	queue	typing
0	-	ϕ	what...	
1–3	skip	ϕ	capital...	
4	$\mathsf{sh}_{\text{capital}}$	**capital**:st→ct	of...	
7	$\mathsf{sh}_{\text{largest}}$	**capital**:st→ct **argmax** : ('a→t)→('a→i)→'a	state...	
8	$\mathsf{sh}_{\text{state}}$	**capital**:st→ct **argmax**:('a→t)→('a→i)→'a **state**:st→t	by...	
9	$\mathsf{re}_{\curvearrowright}$	**capital**:st→ct (**argmax state**):(st→i)→st	by...	binding: 'a = st
11	$\mathsf{sh}_{\text{area}}$	**capital**:st→ct (**argmax state**):(st→i)→st **size**:lo→i	?	
12	$\mathsf{re}_{\curvearrowright}$	**capital**:st→ct (**argmax state size**):st	?	st <: lo ⇒ (lo→i) <: (st→i)
13	$\mathsf{re}_{\curvearrowright}$	(**capital** (**argmax state size**)):ct	?	

Longer-Term Vision

efficiently analyze and generate sequences with hierarchical structures:
natural language, RNA/proteins, programming languages, music, etc.

linear-time search algorithms

structured prediction with deep learning

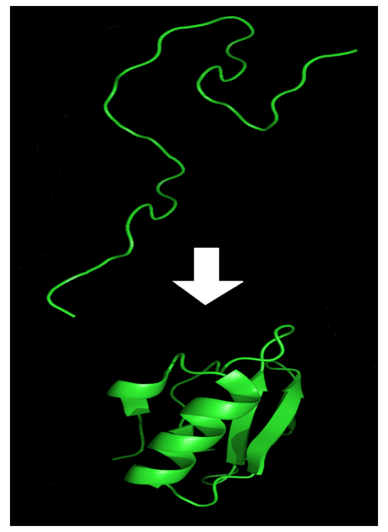
grammar formalisms (context-free & beyond)

the man bit the dog

那 人 咬 了 狗

NL translation

real-time accompaniment



protein folding

self.plural is an lambda function with an argument *n*, which returns result of boolean expression *n* not equal to 1

```
self.plural = lambda n: int(n!=1)
```

NL <=> PL translation

5-Year Vision in Natural Language Processing

5-Year Vision in Natural Language Processing

- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

5-Year Vision in Natural Language Processing

- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

- helping the language-impaired with incremental parsing and simultaneous MT
  - simultaneous English <=> ASL translation; intelligent input system

5-Year Vision in Natural Language Processing

- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

- helping the language-impaired with incremental parsing and simultaneous MT
  - simultaneous English <=> ASL translation; intelligent input system
- online grammatical error correction; automatic or computer-aided writing

5-Year Vision in Natural Language Processing

- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

- helping the language-impaired with incremental parsing and simultaneous MT
  - simultaneous English <=> ASL translation; intelligent input system
- online grammatical error correction; automatic or computer-aided writing
- incremental semantic parsing & code generation: NL=>PL (e.g. SQL)
  - also the inverse problem of PL=>NL translation (comment generation)

5-Year Vision in Natural Language Processing

- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

- helping the language-impaired with incremental parsing and simultaneous MT
  - simultaneous English <=> ASL translation; intelligent input system
- online grammatical error correction; automatic or computer-aided writing
- incremental semantic parsing & code generation: NL=>PL (e.g. SQL)
  - also the inverse problem of PL=>NL translation (comment generation)
- how do incremental predictive parsers compare with psycholinguistics data?

5-Year Vision in Natural Language Processing

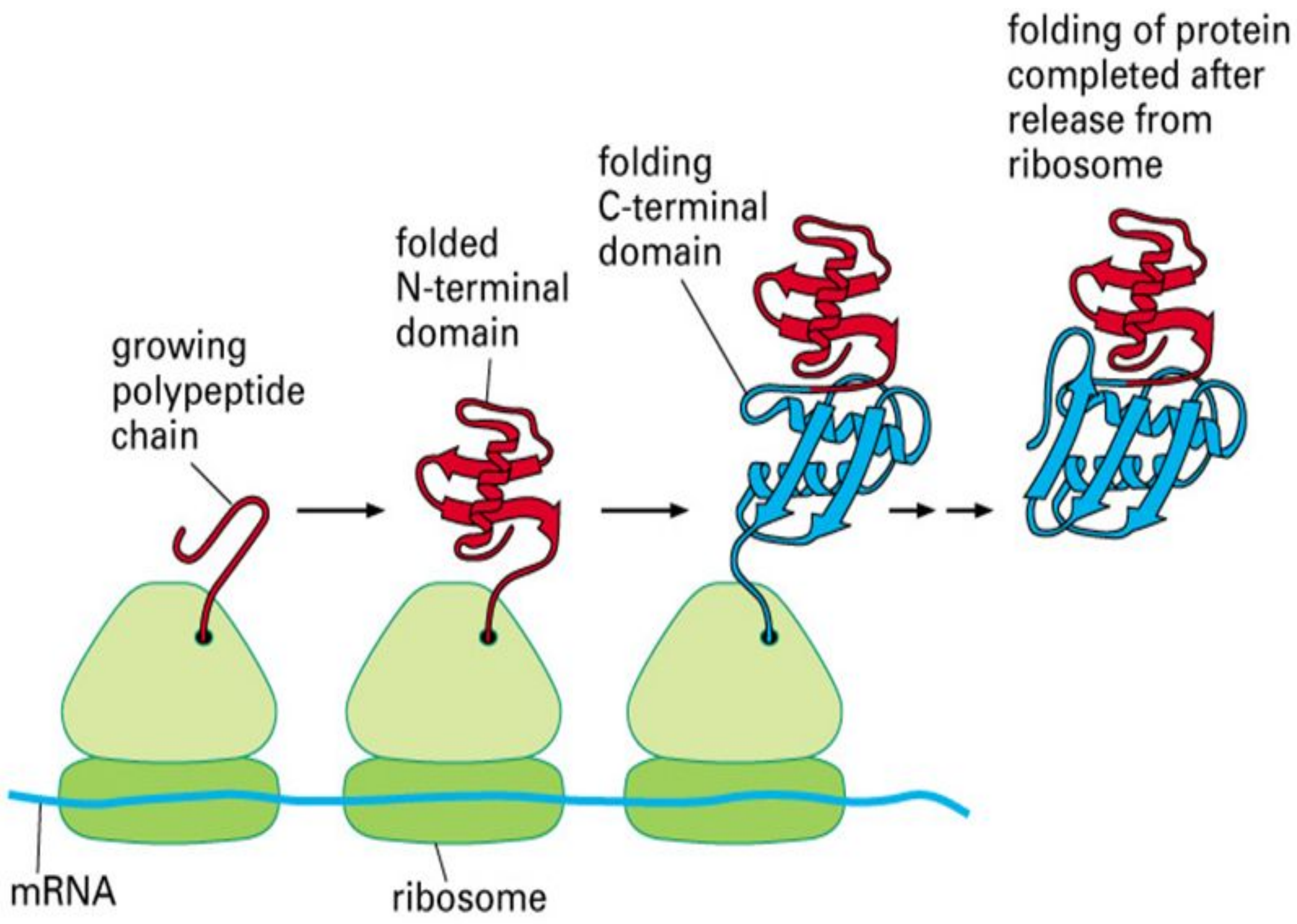
helping people communicate across linguistic and accessibility barriers

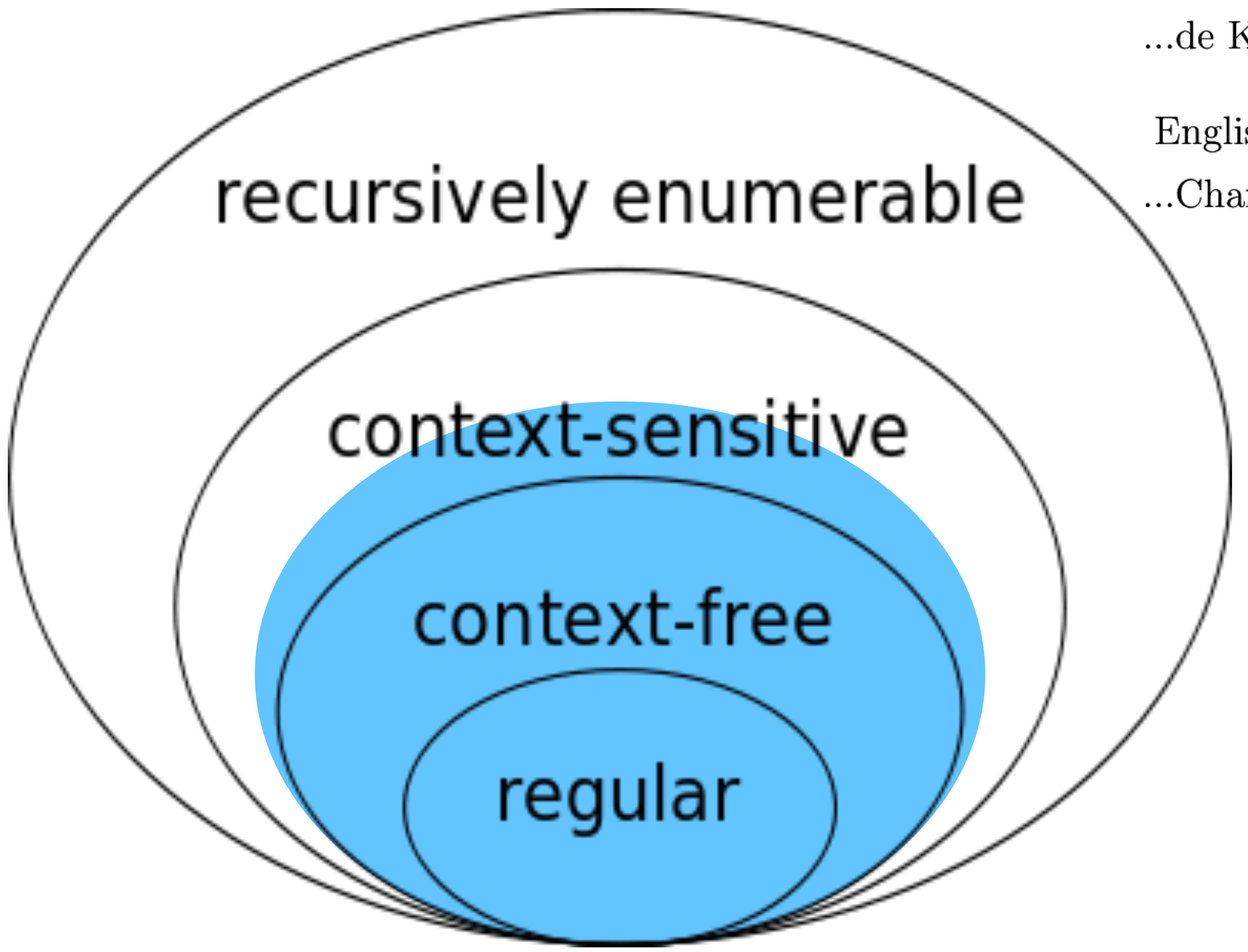
- simultaneous translation: from speech-to-text to speech-to-speech
  - incremental text-to-speech synthesis (*language production is also incremental!*)
  - incremental predictive parsing on the source side (improve reordering)
  - incremental predictive parsing on the target side (improve prosody)

- helping the language-impaired with incremental parsing and simultaneous MT
  - simultaneous English <=> ASL translation; intelligent input system
- online grammatical error correction; automatic or computer-aided writing
- incremental semantic parsing & code generation: NL=>PL (e.g. SQL)
  - also the inverse problem of PL=>NL translation (comment generation)
- how do incremental predictive parsers compare with psycholinguistics data?

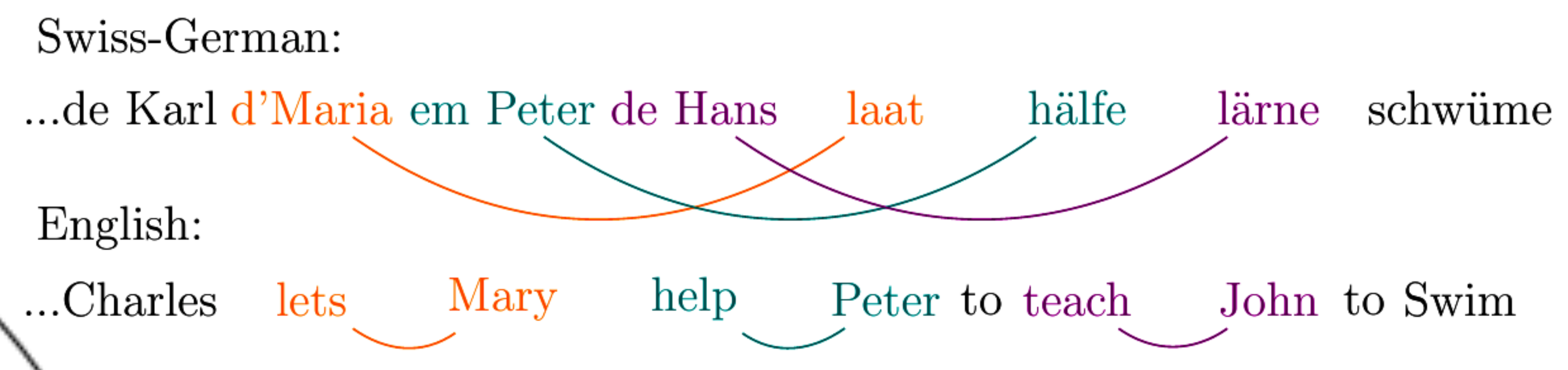
5-Year Vision in Computational Biology

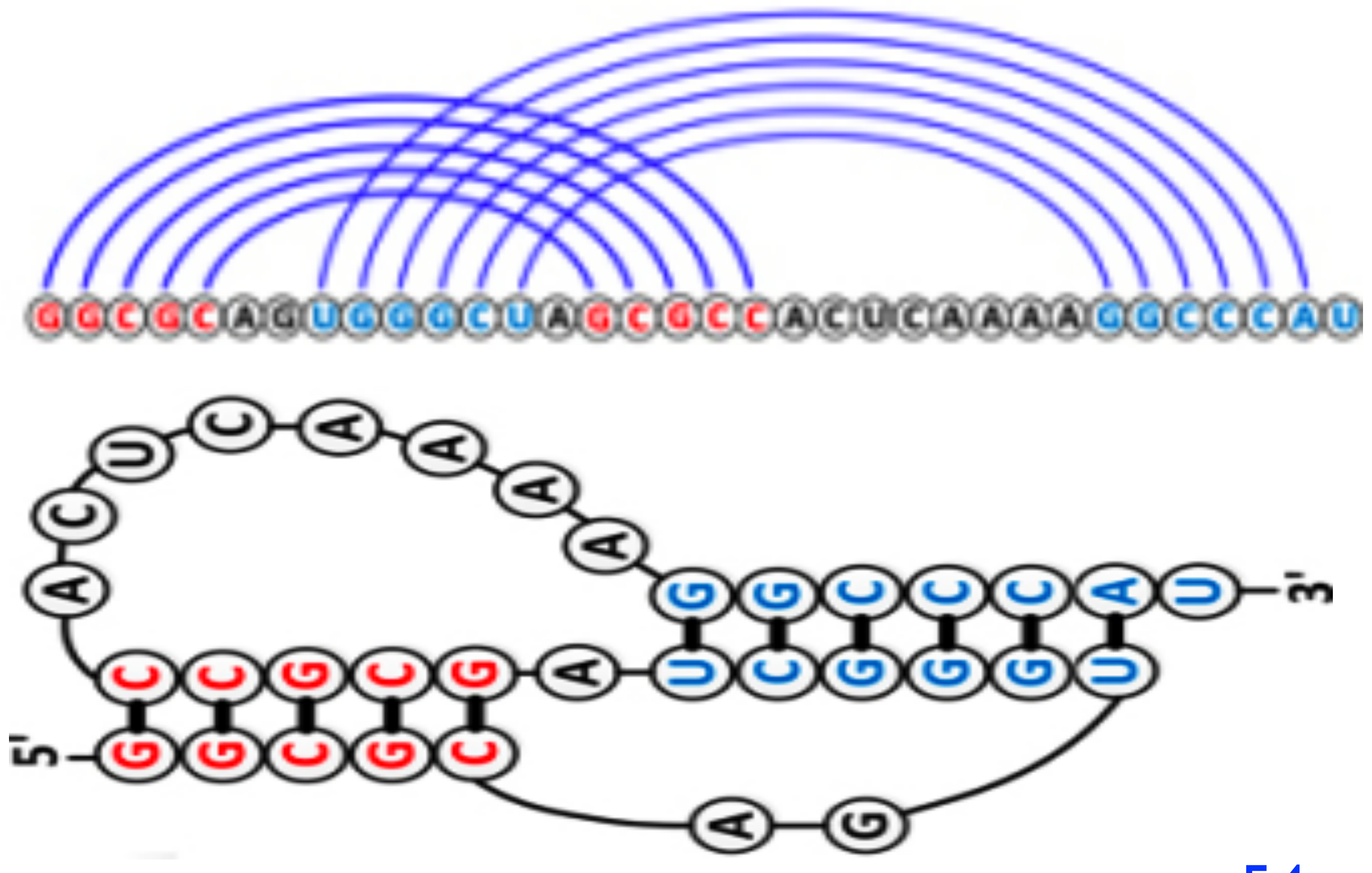
- linear-time incremental folding: from RNA to protein structure prediction
- predicting crossing structures in RNAs/proteins: use linear-time parsing and *mildly* context-sensitive grammars (polynomial-time parsable)
- how does our beam search compare to real incremental folding in nature?





Chomsky Hierarchy

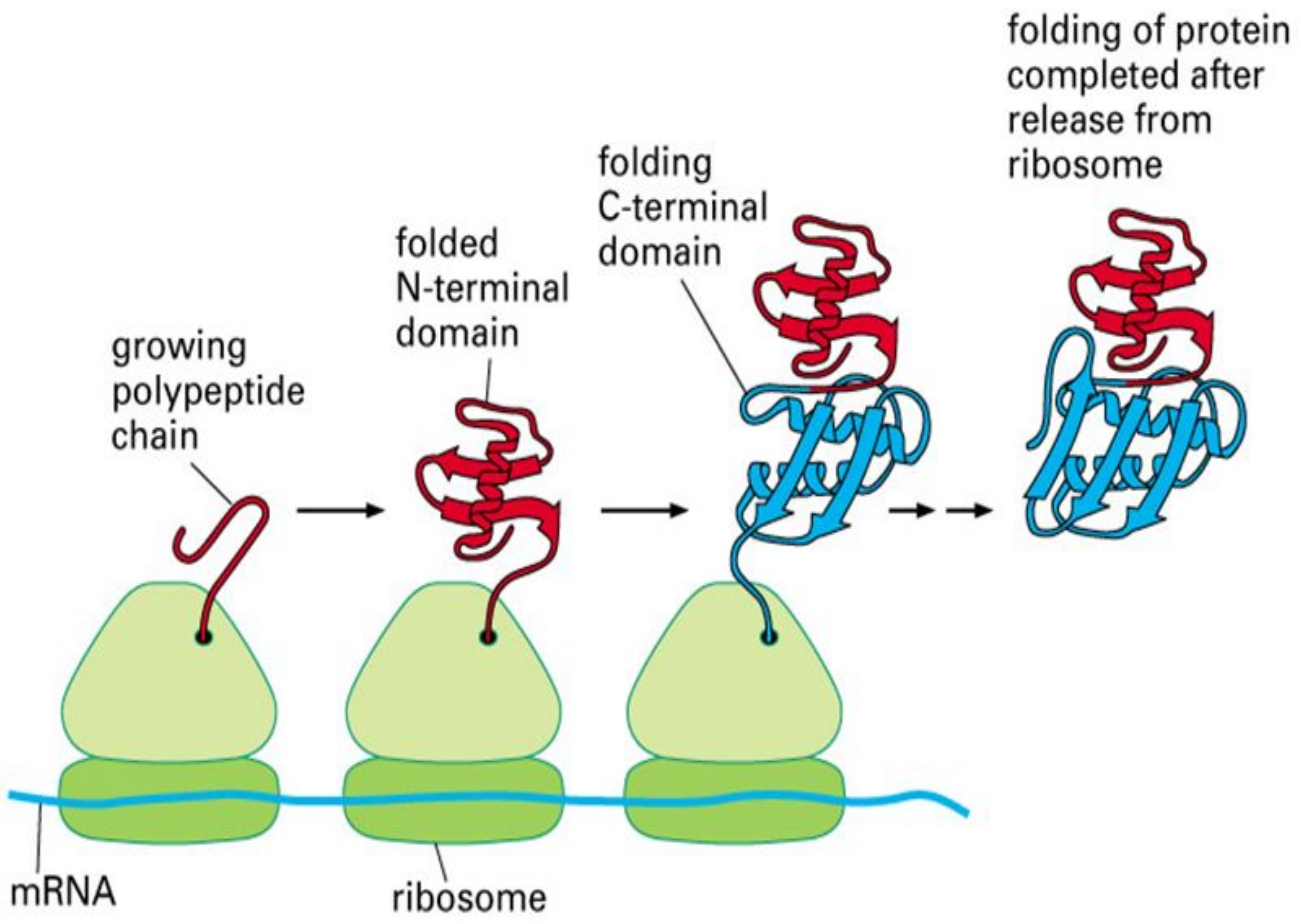


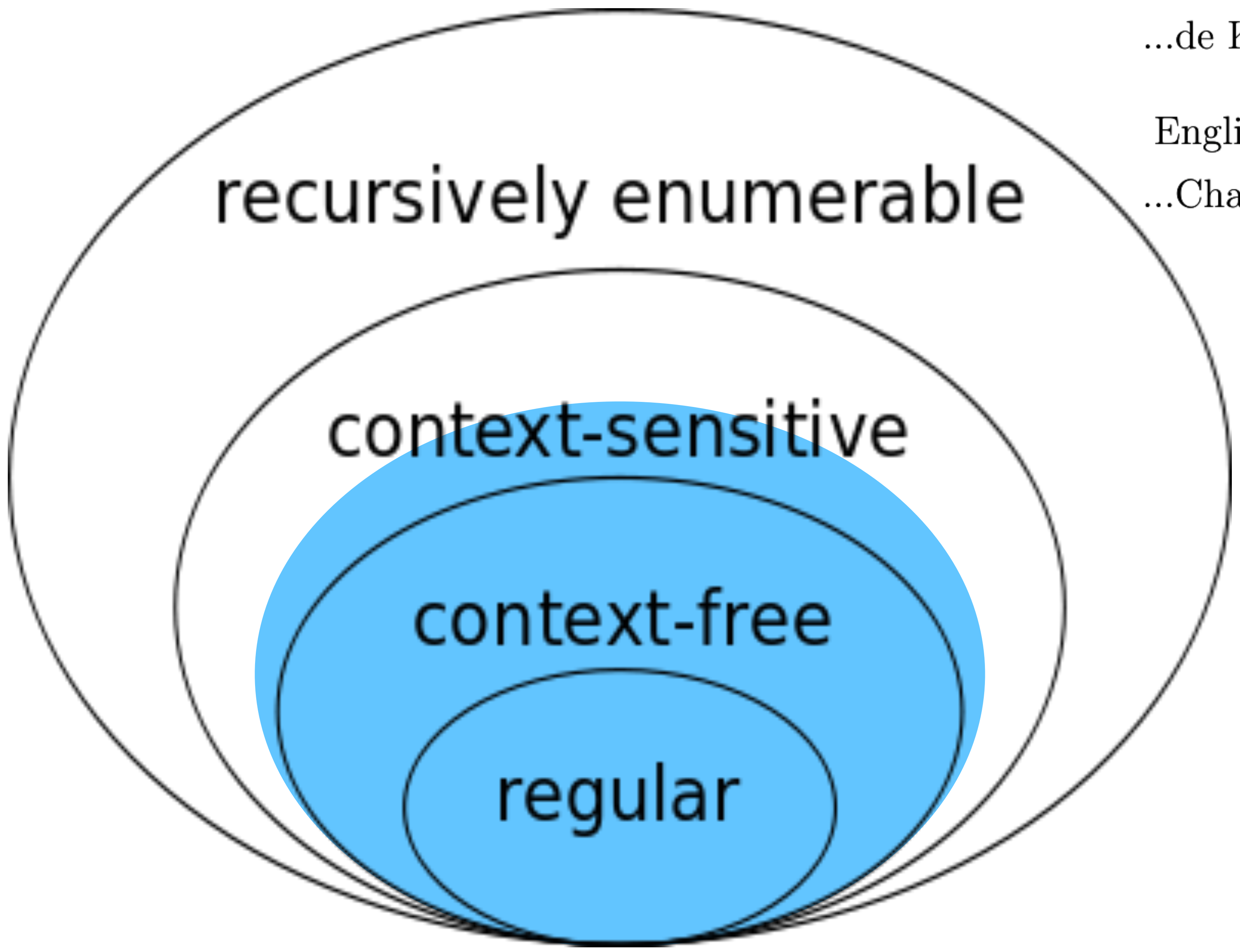


5-Year Vision in

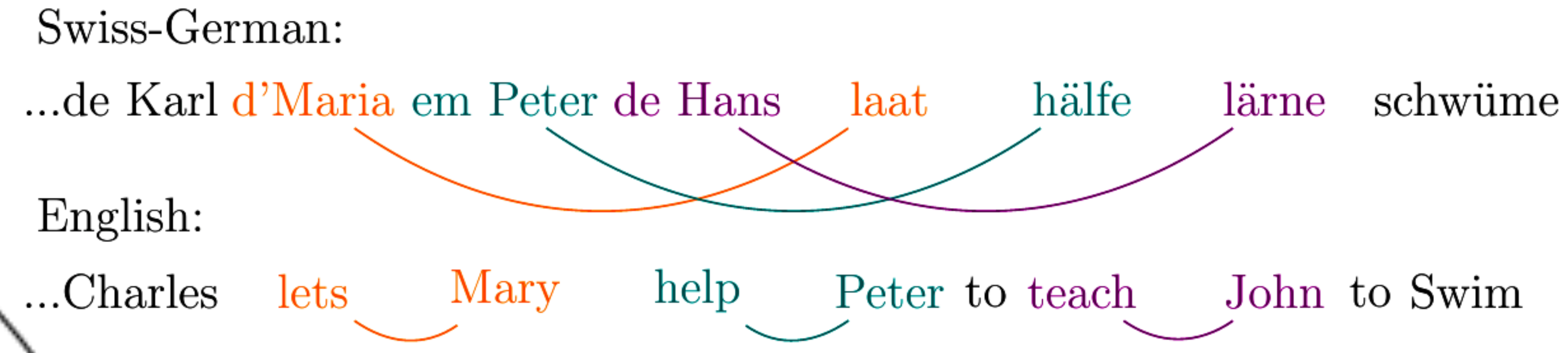
reestablishing the forgotten link between computational linguistics and structural biology

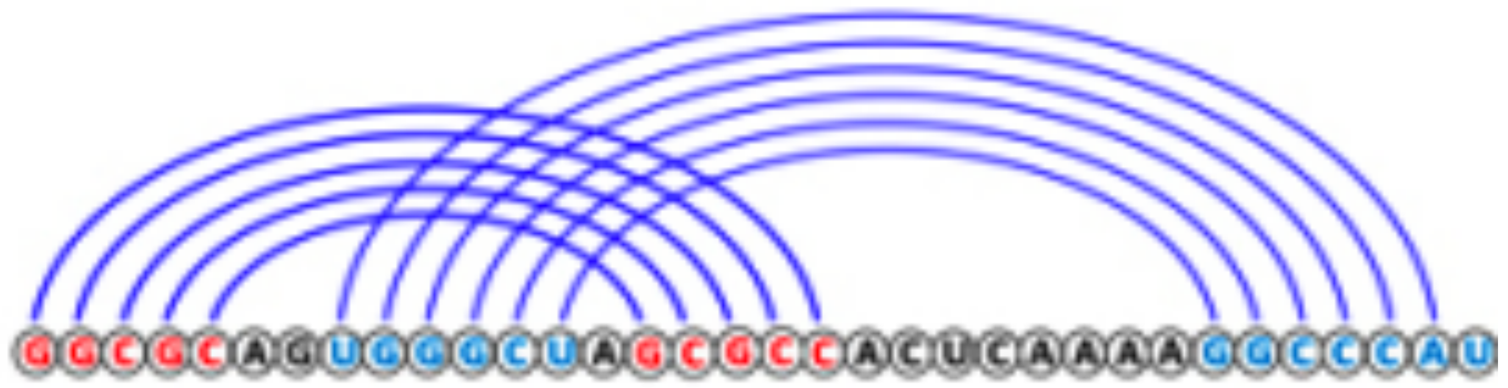
- linear-time incremental folding: from RNA to protein structure prediction
- predicting crossing structures in RNAs/proteins: use linear-time parsing and *mildly* context-sensitive grammars (polynomial-time parsable)
- how does our beam search compare to real incremental folding in nature?

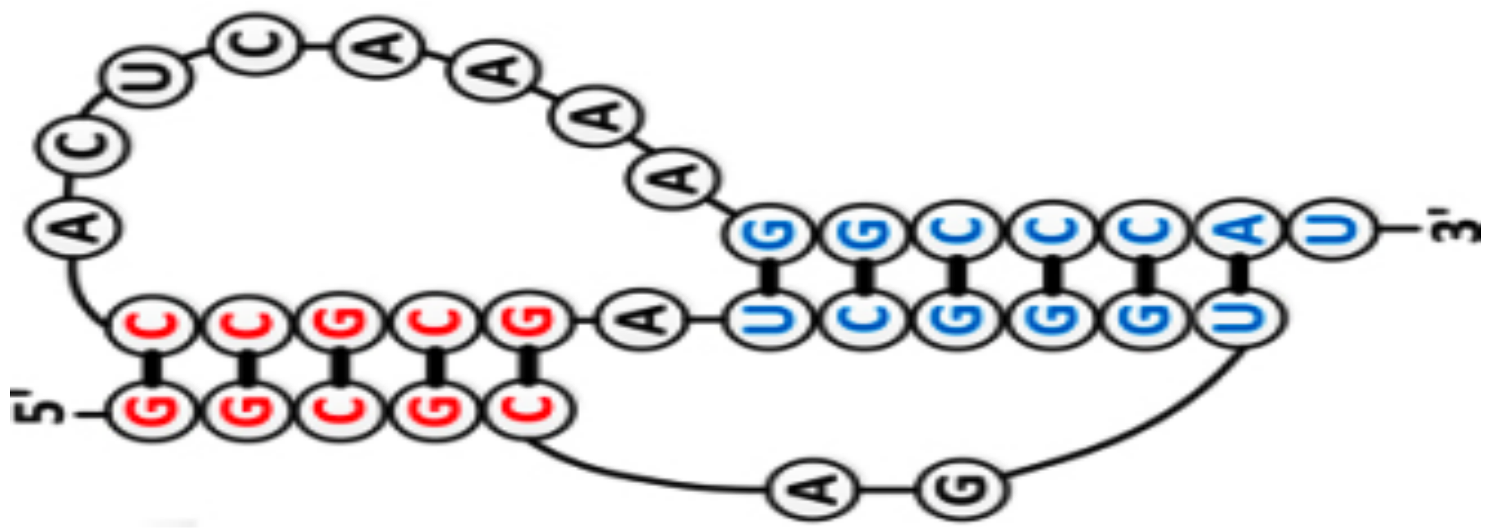




Chomsky Hierarchy







非常 感谢 您 来 听 我 的 演讲

Thank you very much for listening to my speech

Bush met Putin in Moscow	I eat sushi with tuna from Japan	GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA
source language sentence	natural language sentence	RNA sequence

Baidu Research

Oregon State University

非常 感谢 您　来 听 我　的 演讲

Thank you very much for listening to my speech

Happy Chinese New Year!

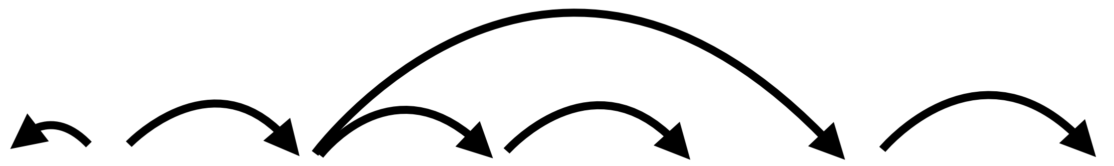
target-language sequence

布什在莫斯科与普京会晤

Bush met Putin in Moscow

source language sentence

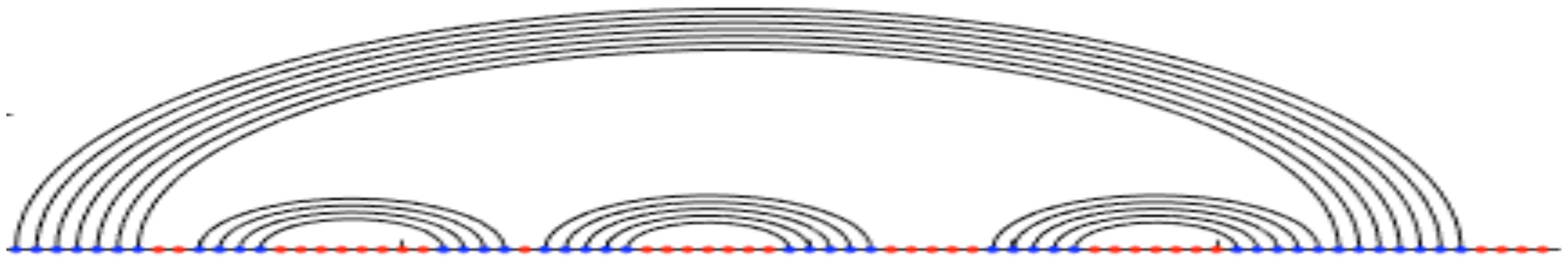
syntactic structure



I eat sushi with tuna from Japan

natural language sentence

secondary structure



GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

RNA sequence

Baidu Research

Oregon State University

Backup Slide

Baidu Research

Chinese input:

Pinyin:

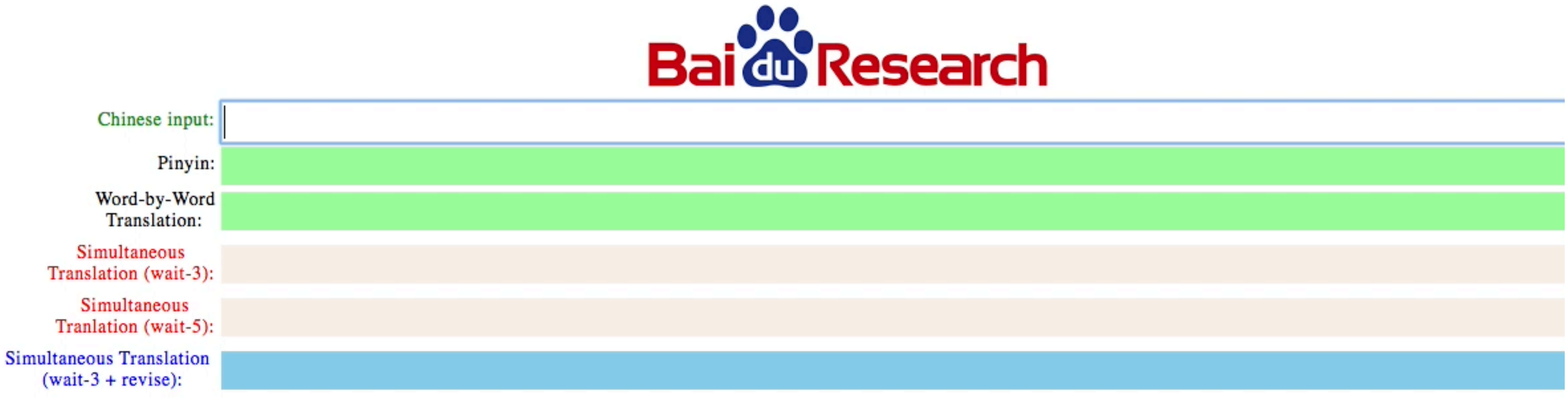
Word-by-Word Translation:

Simultaneous Translation (wait-3):

Simultaneous Tranlation (wait-5):

Simultaneous Translation (wait-3 + revise):

Backup Slide



Translation with Noisy Speech Input

- neural MT is fragile, and automatic speech recognition output is noisy
- our work (Liu et al, ArXiv 2018): Robust Neural MT using phonetic information

Clean Input	目前已发现有109人死亡, 另有57人获救
Output of Transformer	at present, 109 people have been found dead and 57 have been rescued
Noisy Input	目前已发现又109人死亡, 另有57人获救
Output of Transformer	the hpv has been found dead so far and 57 have been saved
Output of Our Method	so far, 109 people have been found dead and 57 others have been rescued

Table 1: The translation results on Mandarin sentences without and with homophone noises. The word '有' (yǒu, "have") in clean input is replaced by one of its homophone, '又' (yòu, "again"), to form a noisy input. This seemingly minor change completely fools the Transformer to generate something irrelvant ("hpv"). Our method, by contrast, is very robust to homophone noises thanks to phonetic information.