

Bringing together grammar and deep learning: models and targeted evaluation

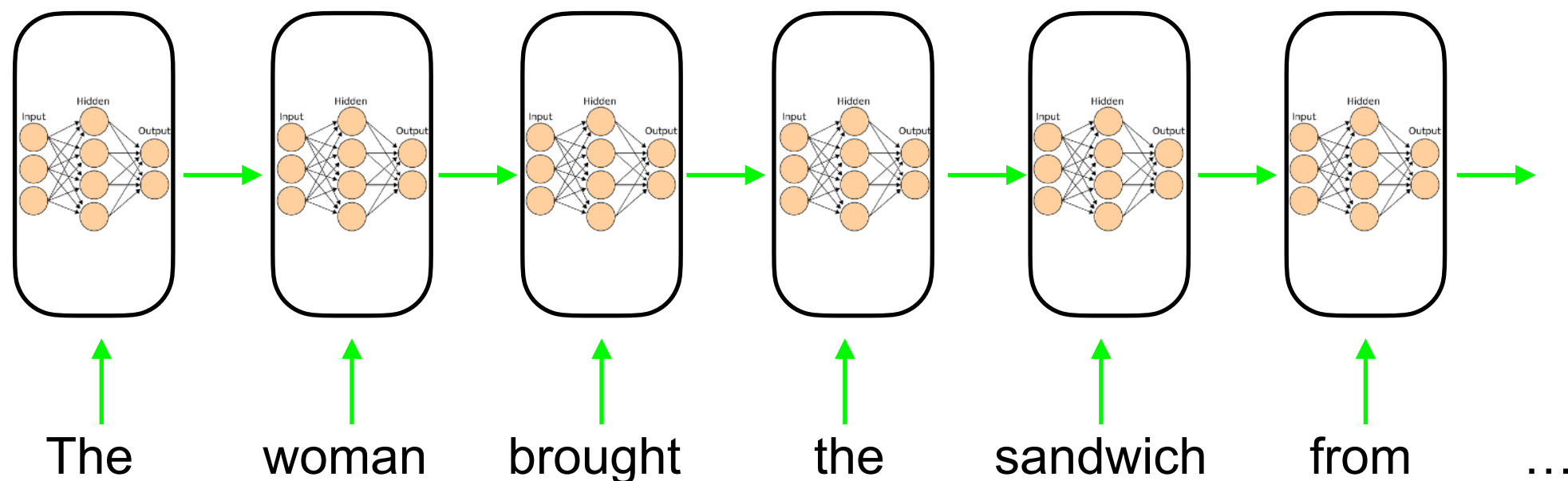
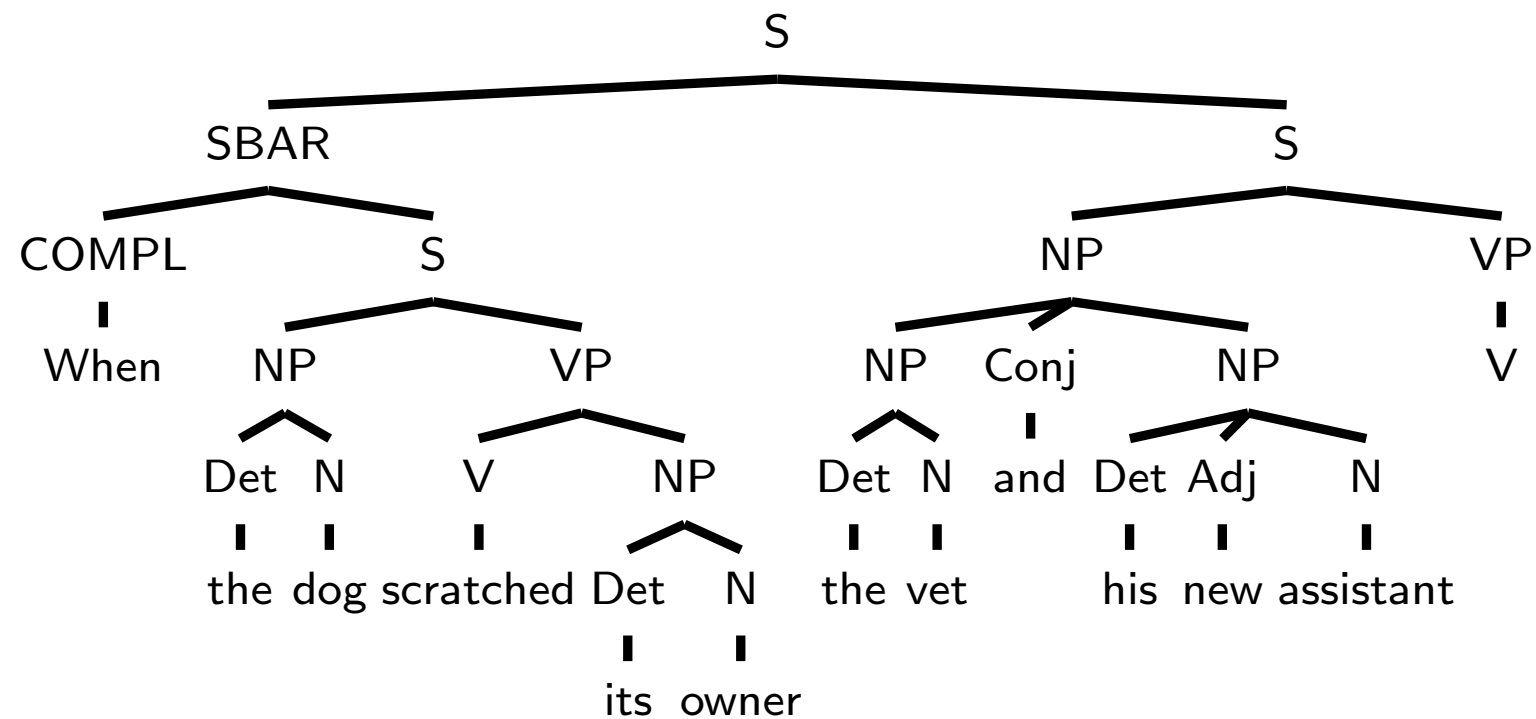


Roger Levy
9.19: Computational Psycholinguistics
10 November 2021

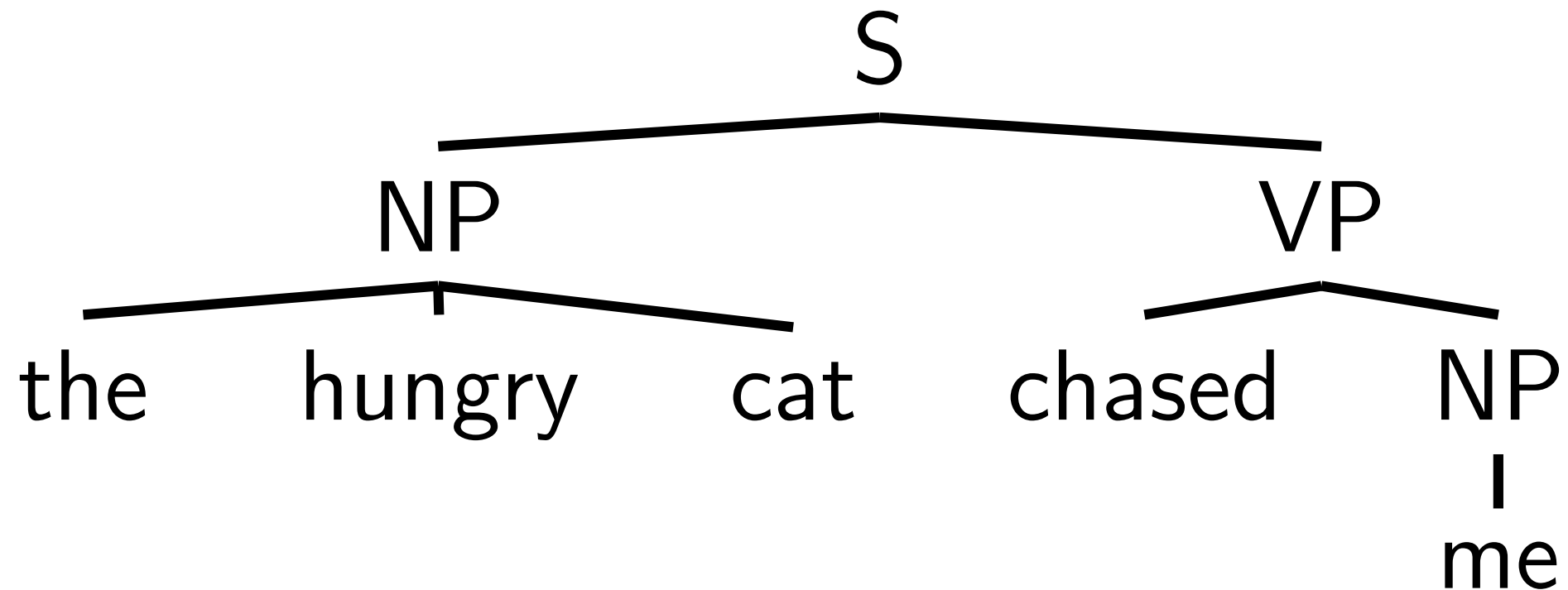
Agenda for today

- Combining symbolic grammar and neural generalization
- Controlled tests for syntactic generalization:
 - Subordination
 - Garden-pathing

Grammar and deep learning



Sequence representations of trees



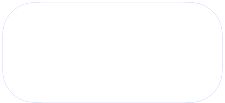
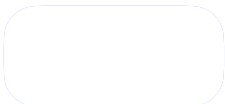
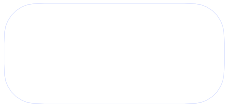
(S (NP the hungry cat) (VP chased (NP me)))

(S (NP the hungry cat) (VP chased (NP me)))

This can be seen as an *action sequence*!

Action	Meaning	String gloss
NT(X)	Push a new open non-terminal on top of the stack	(X
Gen(w)	Generate word w as a terminal node and put it on top of the stack (as a closed node)	w
REDUCE	Pop closed nodes $N_{1...i-1}$ from the top of the stack until encountering open node N_i ; close N_i)
END	Finish parsing (iff the sole stack element is a closed S)	n/a

(S (NP the hungry cat) (VP chased (NP me)))



(S (NP the hungry cat) (VP chased (NP me)))

Action

Stack

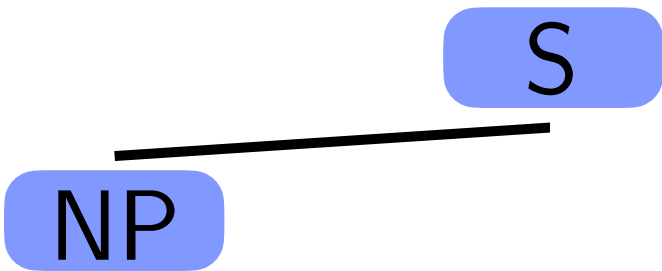
(S (NP the hungry cat) (VP chased (NP me)))

S

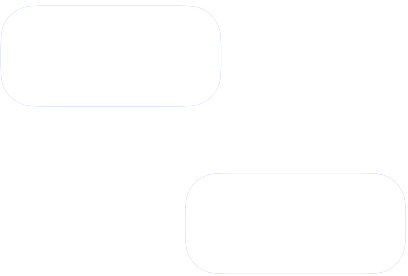
Action
NT(S)

Stack
(S

(S (NP the hungry cat) (VP chased (NP me)))



Action	Stack
NT(S)	(S
NT(NP)	(S (NP



(S (NP the hungry cat) (VP chased (NP me)))

Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

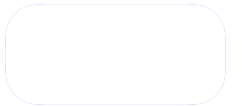
Gen(the)

(S | (NP | the

the

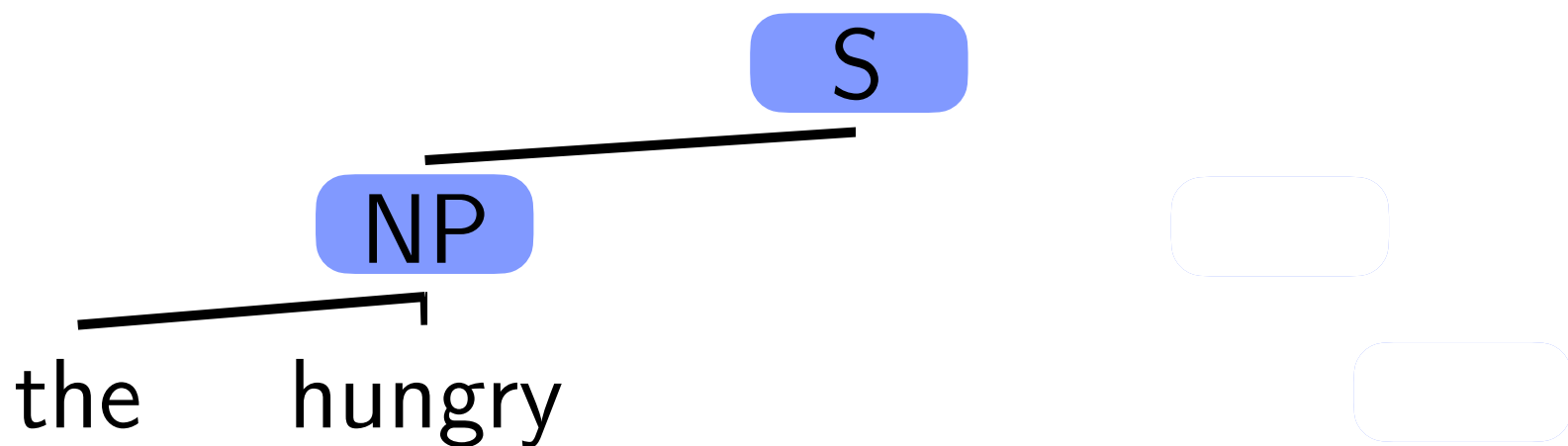
NP

S



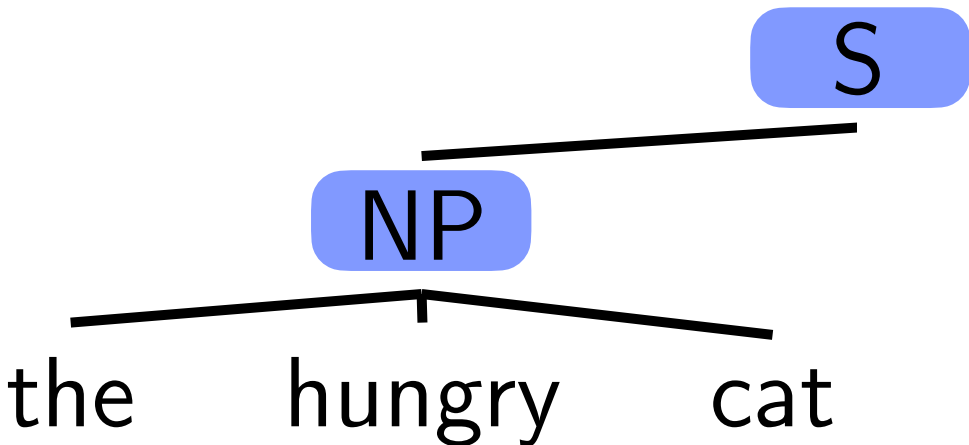
(S (NP the hungry cat) (VP chased (NP me)))

Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry

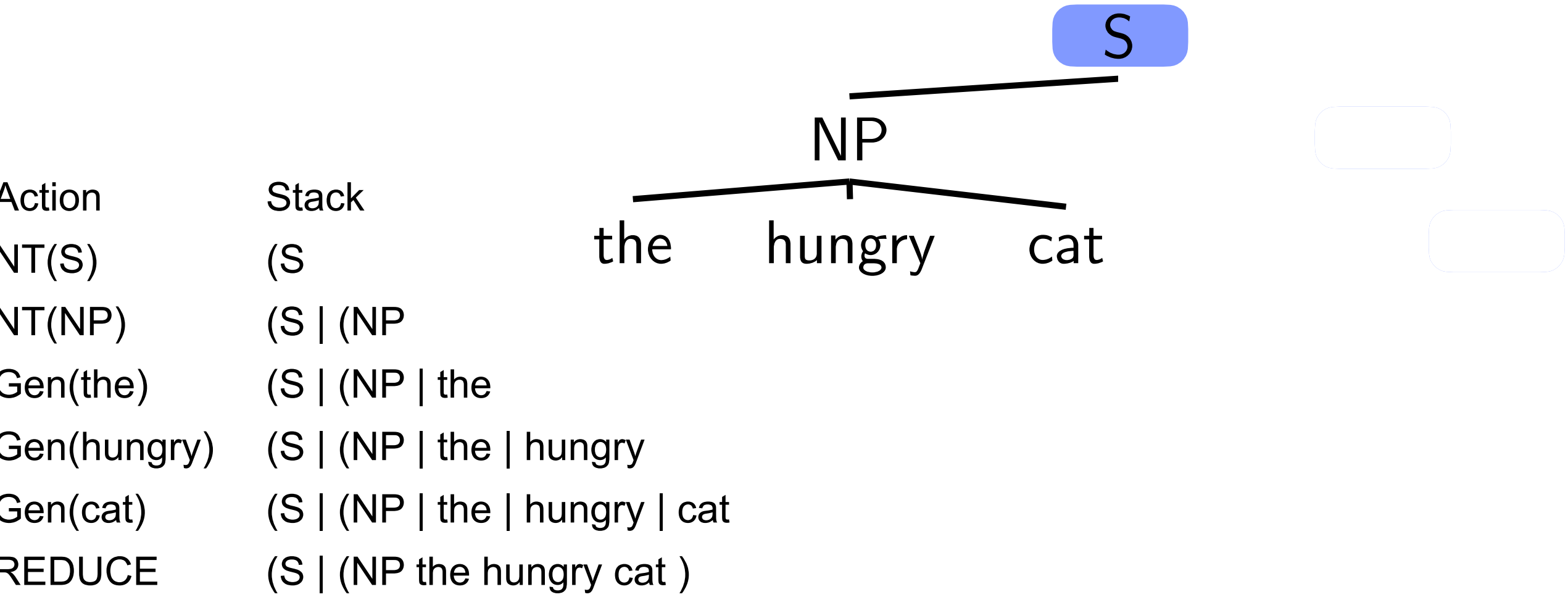


(S (NP the hungry cat) (VP chased (NP me)))

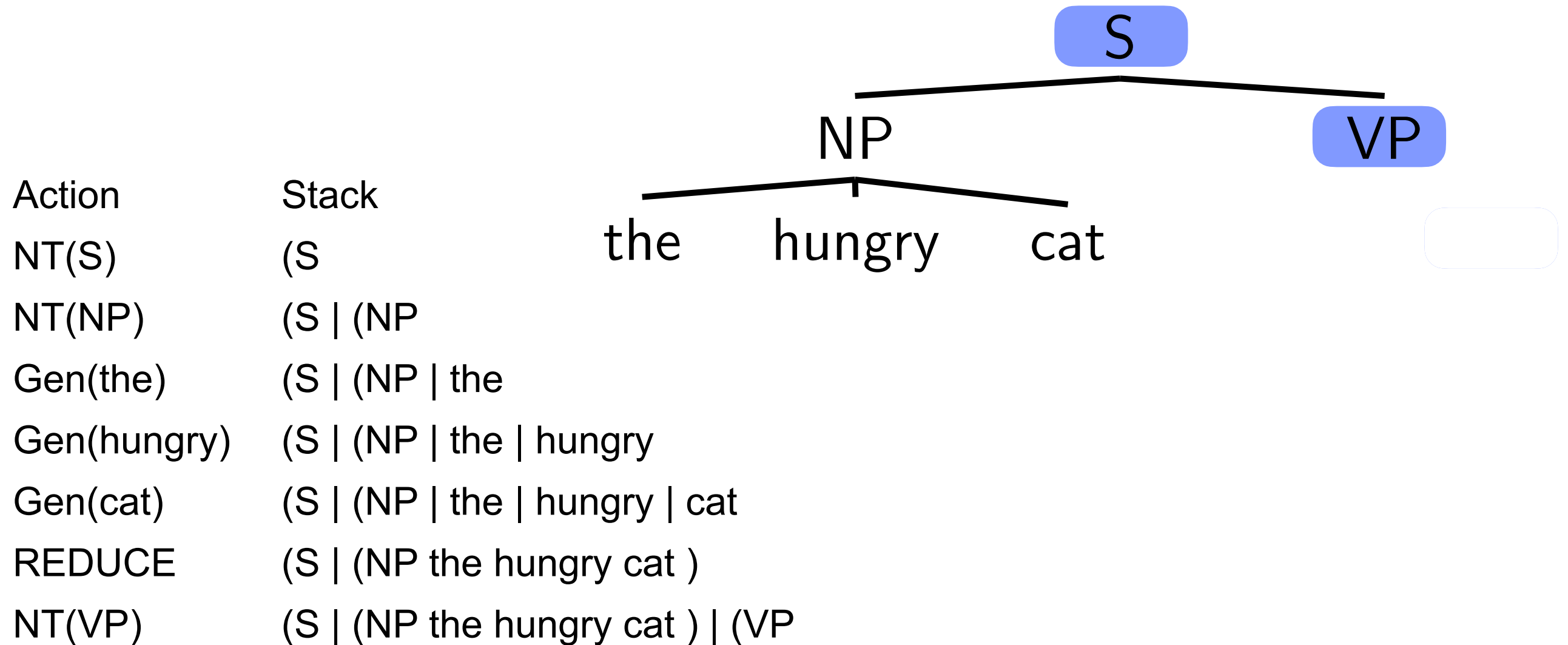
Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat



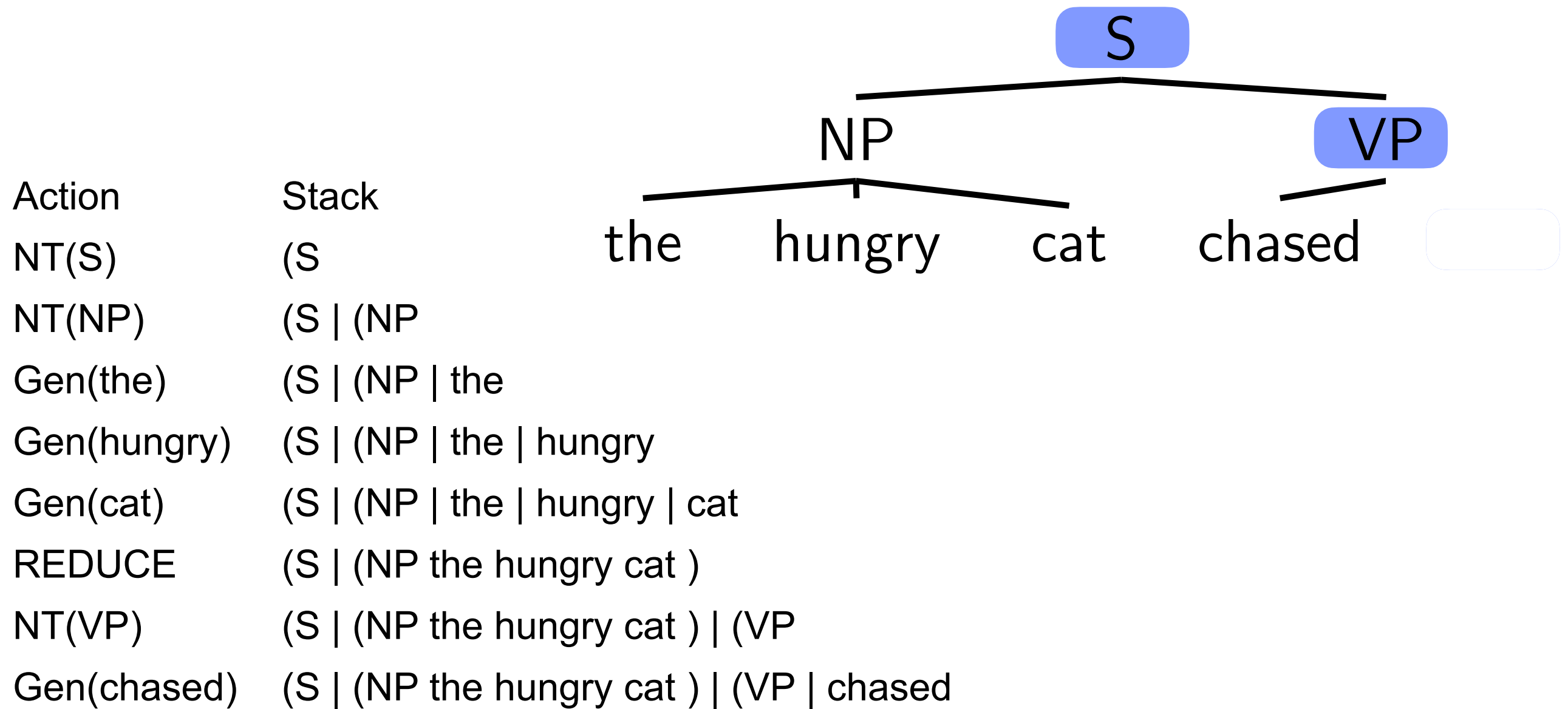
(S (NP the hungry cat) (VP chased (NP me)))



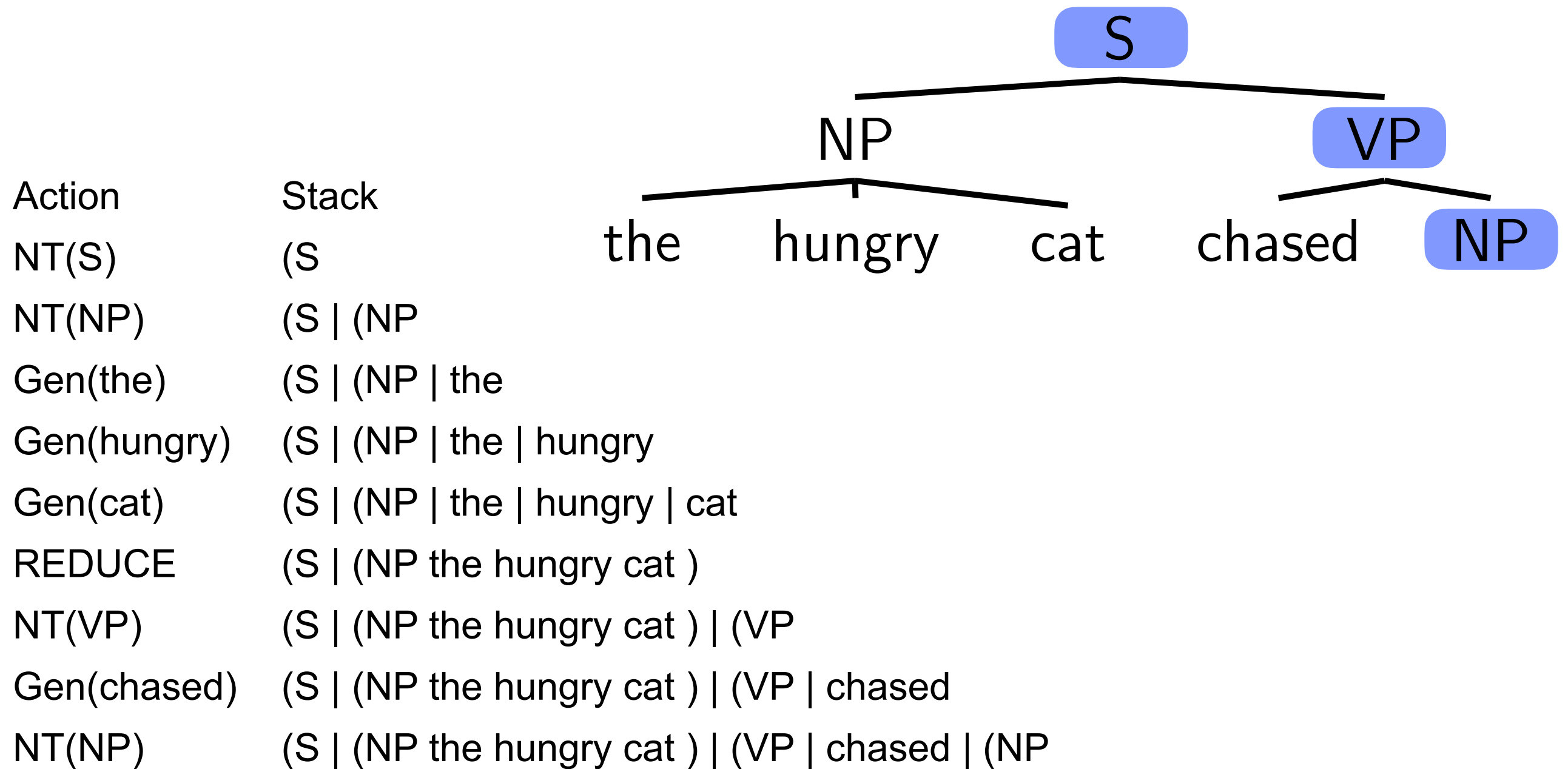
(S (NP the hungry cat) (VP chased (NP me)))



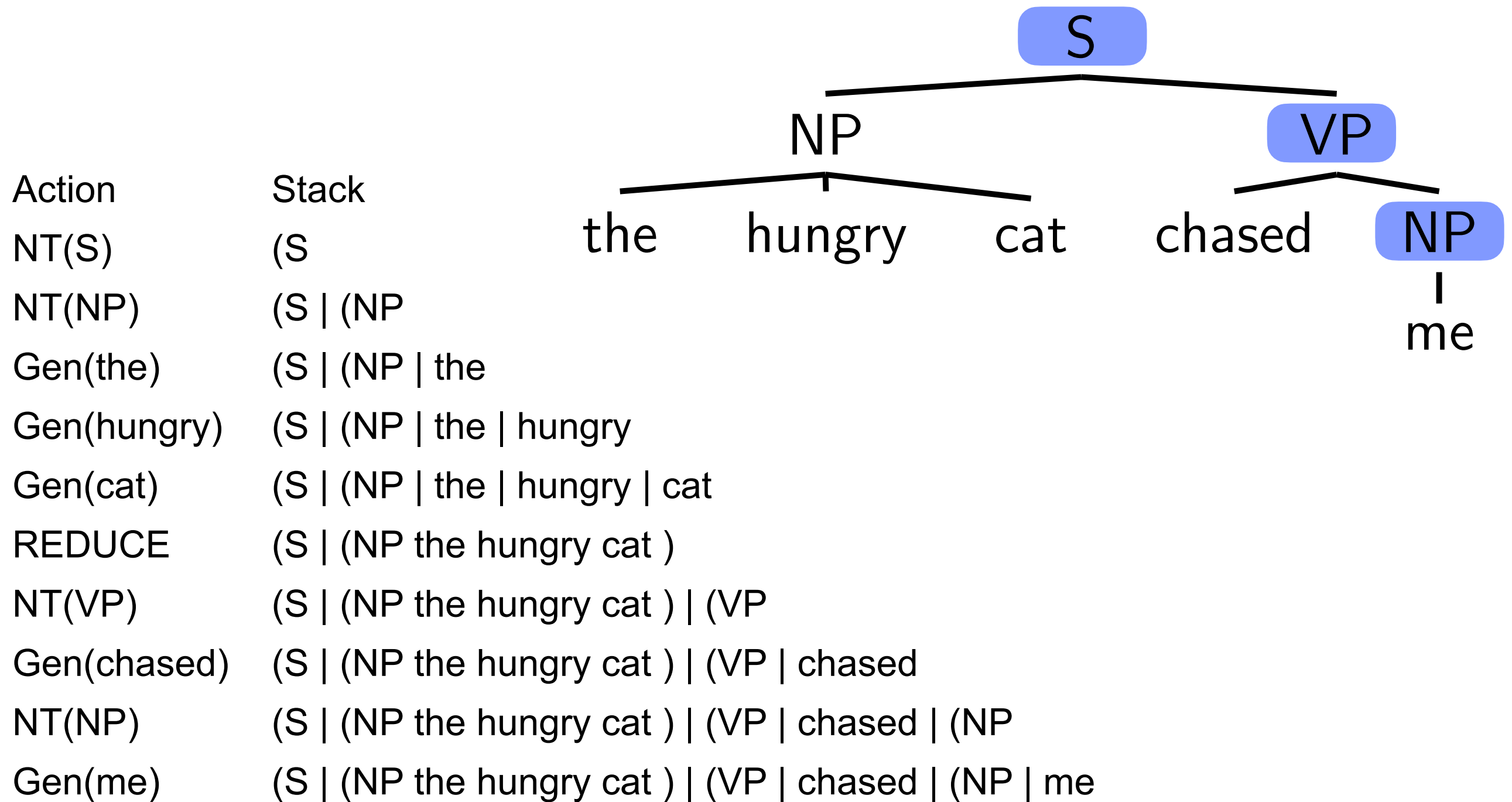
(S (NP the hungry cat) (VP chased (NP me)))



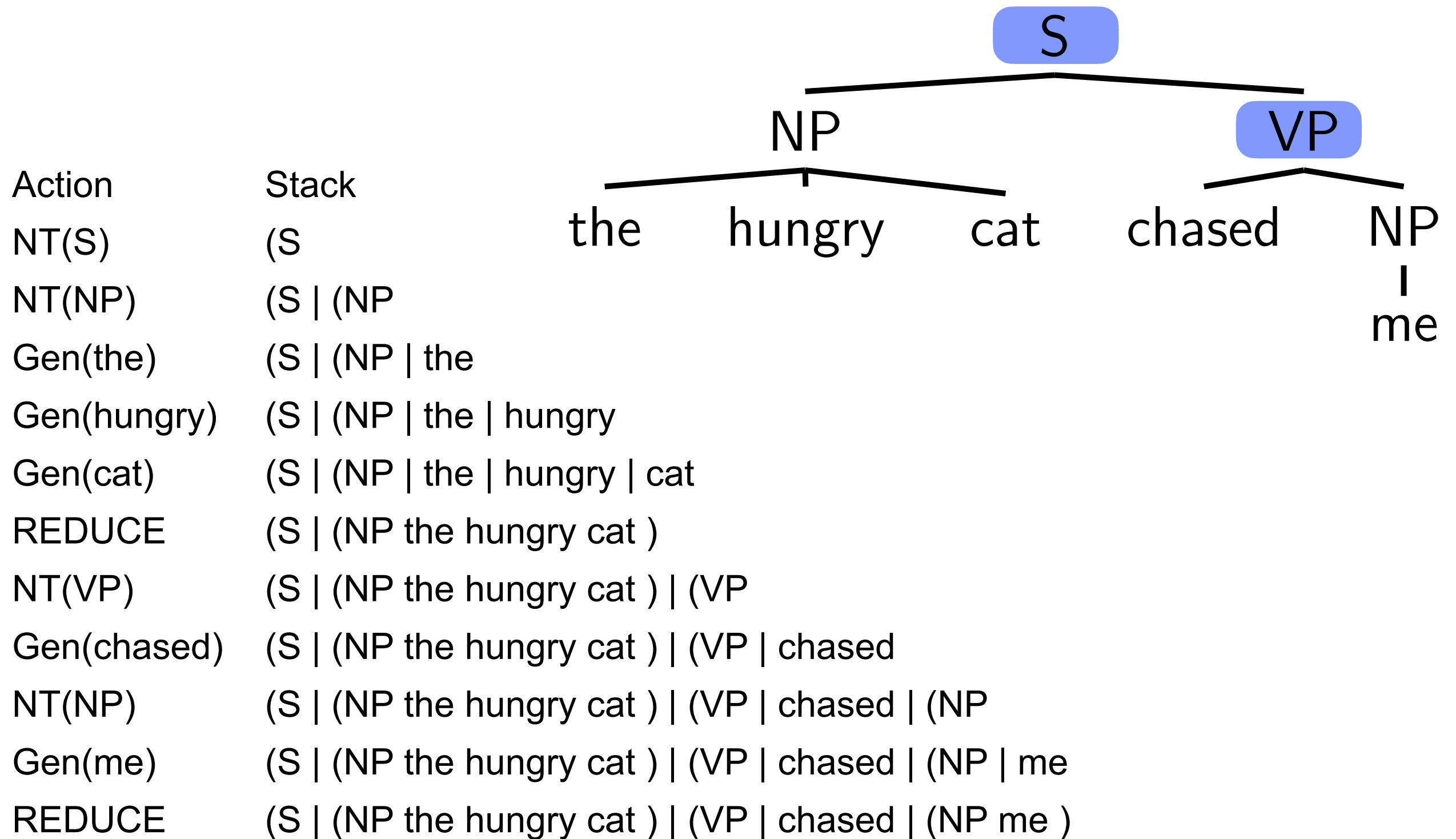
(S (NP the hungry cat) (VP chased (NP me)))



(S (NP the hungry cat) (VP chased (NP me)))



(S (NP the hungry cat) (VP chased (NP me)))



(S (NP the hungry cat) (VP chased (NP me)))

		<pre> graph TD S[S] --- NP1[NP] S --- VP[VP] NP1 --- the[the] NP1 --- hungry[hungry] NP1 --- cat[cat] VP --- chased[chased] VP --- NP2[NP] NP2 --- me[me] </pre>				
Action	Stack					
NT(S)	(S	the	hungry	cat	chased	NP
NT(NP)	(S (NP					
Gen(the)	(S (NP the					me
Gen(hungry)	(S (NP the hungry					
Gen(cat)	(S (NP the hungry cat					
REDUCE	(S (NP the hungry cat)					
NT(VP)	(S (NP the hungry cat) (VP					
Gen(chased)	(S (NP the hungry cat) (VP chased					
NT(NP)	(S (NP the hungry cat) (VP chased (NP					
Gen(me)	(S (NP the hungry cat) (VP chased (NP me					
REDUCE	(S (NP the hungry cat) (VP chased (NP me)					
REDUCE	(S (NP the hungry cat) (VP chased (NP me))					

(S (NP the hungry cat) (VP chased (NP me)))

		<pre> graph TD S --> NP1[NP] S --> VP[VP] NP1 --> the[the] NP1 --> hungry[hungry] NP1 --> cat[cat] VP --> chased[chased] VP --> NP2[NP] NP2 --> me[me] </pre>				
Action	Stack					
NT(S)	(S	the	hungry	cat	chased	NP
NT(NP)	(S (NP					
Gen(the)	(S (NP the					me
Gen(hungry)	(S (NP the hungry					
Gen(cat)	(S (NP the hungry cat					
REDUCE	(S (NP the hungry cat)					
NT(VP)	(S (NP the hungry cat) (VP					
Gen(chased)	(S (NP the hungry cat) (VP chased					
NT(NP)	(S (NP the hungry cat) (VP chased (NP					
Gen(me)	(S (NP the hungry cat) (VP chased (NP me					
REDUCE	(S (NP the hungry cat) (VP chased (NP me)					
REDUCE	(S (NP the hungry cat) (VP chased (NP me))					
REDUCE	(S (NP the hungry cat) (VP chased (NP me)))					

(S (NP the hungry cat) (VP chased (NP me)))

Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

NT(NP)

(S | (NP the hungry cat) | (VP | chased | (NP

Gen(me)

(S | (NP the hungry cat) | (VP | chased | (NP | me

REDUCE

(S | (NP the hungry cat) | (VP | chased | (NP me)

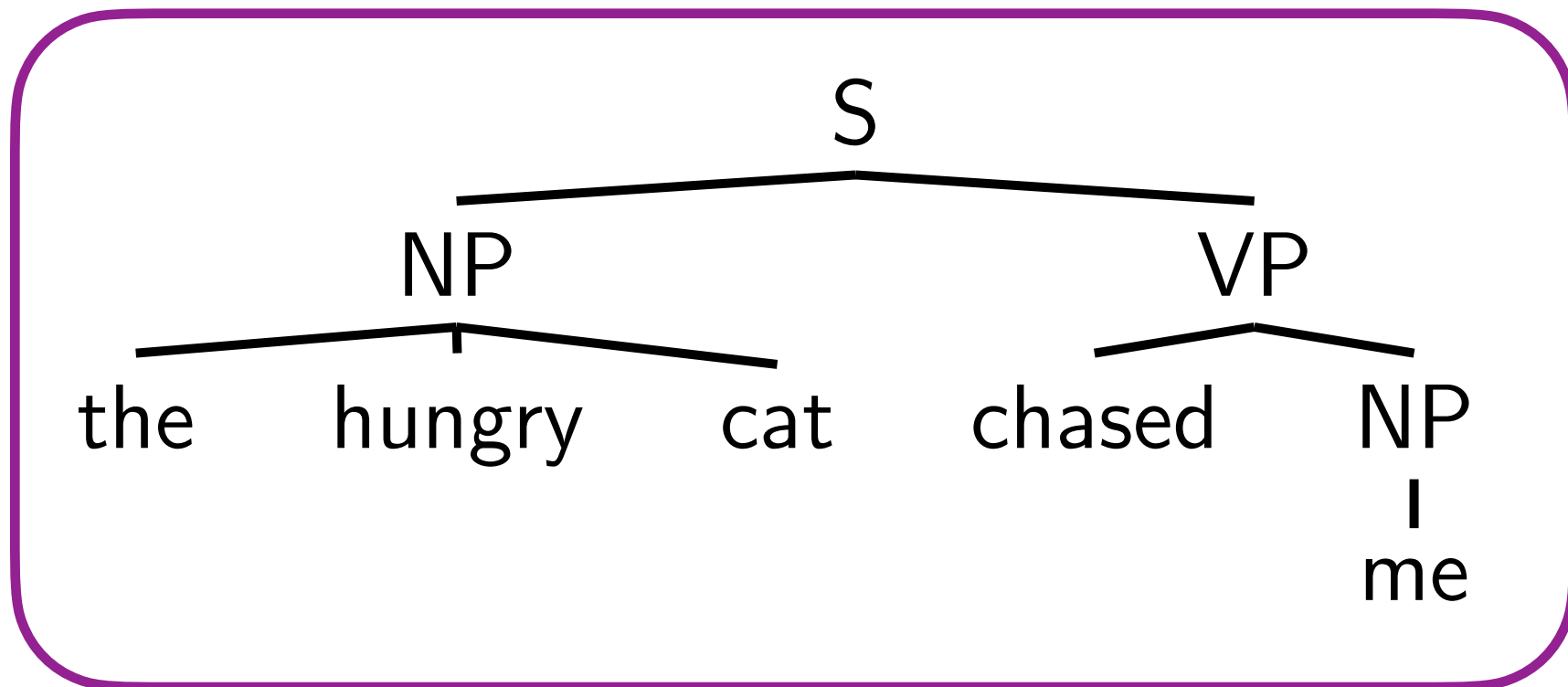
REDUCE

(S | (NP the hungry cat) | (VP chased (NP me))

REDUCE

(S (NP the hungry cat) (VP chased (NP me)))

END



(S (NP the hungry cat) (VP chased (NP me)))

Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

NT(NP)

(S | (NP the hungry cat) | (VP | chased | (NP

Gen(me)

(S | (NP the hungry cat) | (VP | chased | (NP | me

REDUCE

(S | (NP the hungry cat) | (VP | chased | (NP me)

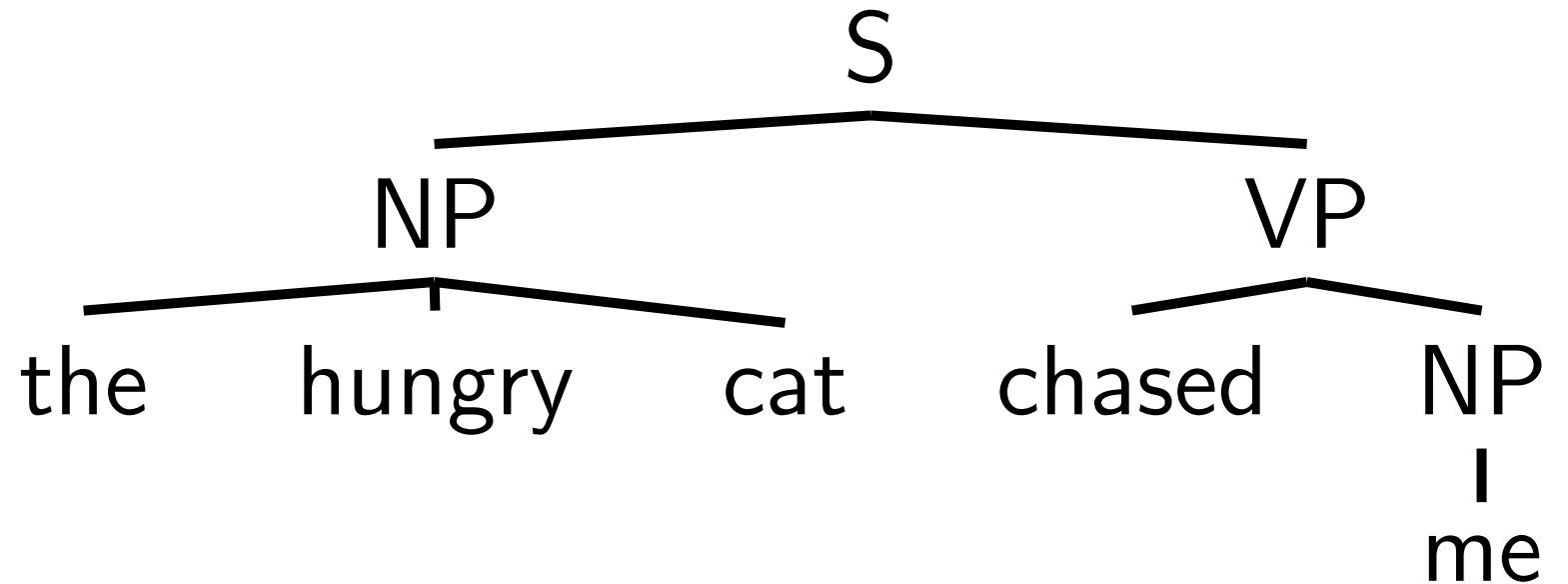
REDUCE

(S | (NP the hungry cat) | (VP chased (NP me))

REDUCE

(S (NP the hungry cat) (VP chased (NP me)))

END



(S (NP the hungry cat) (VP chased (NP me)))

If we put a conditional probability distribution on actions, we have a probabilistic grammar!

Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

NT(NP)

(S | (NP the hungry cat) | (VP | chased | (NP

Gen(me)

(S | (NP the hungry cat) | (VP | chased | (NP | me

REDUCE

(S | (NP the hungry cat) | (VP | chased | (NP me)

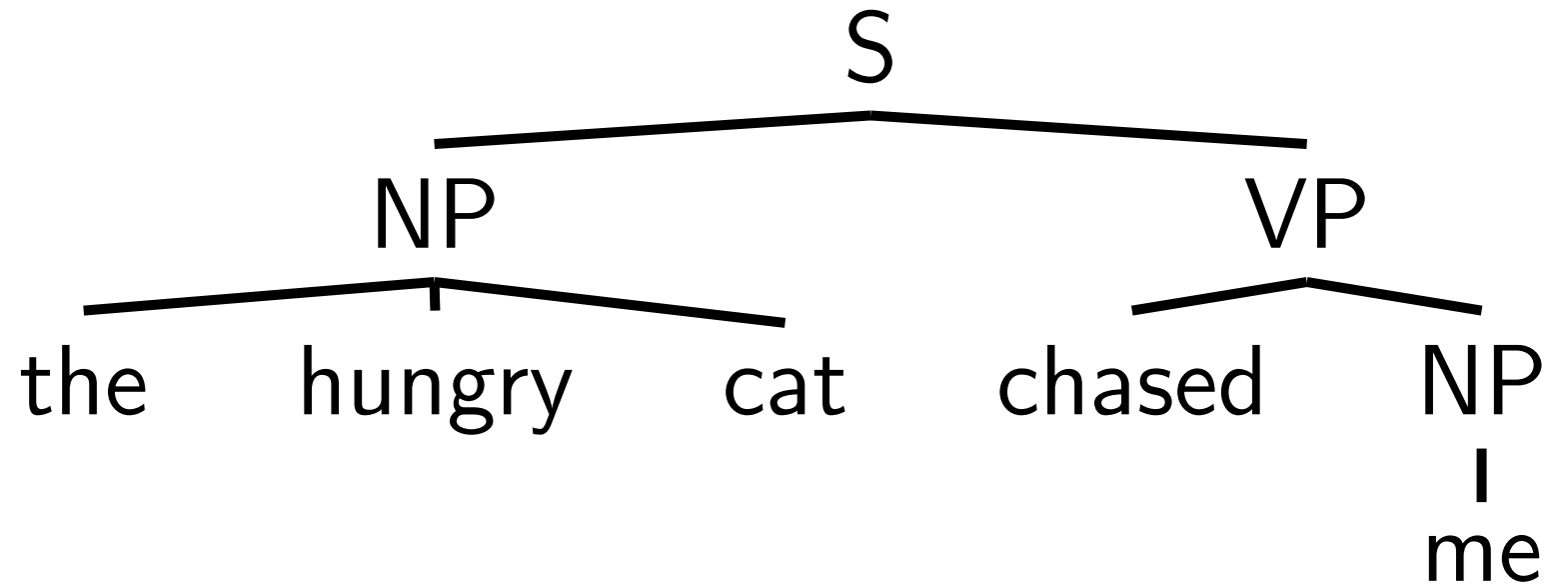
REDUCE

(S | (NP the hungry cat) | (VP chased (NP me))

REDUCE

(S (NP the hungry cat) (VP chased (NP me)))

END



(S (NP the hungry cat) (VP chased (NP me)))

If we put a conditional probability distribution on actions, we have a probabilistic grammar!

Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

NT(NP)

(S | (NP the hungry cat) | (VP | chased | (NP

Gen(me)

(S | (NP the hungry cat) | (VP | chased | (NP | me

REDUCE

(S | (NP the hungry cat) | (VP | chased | (NP me)

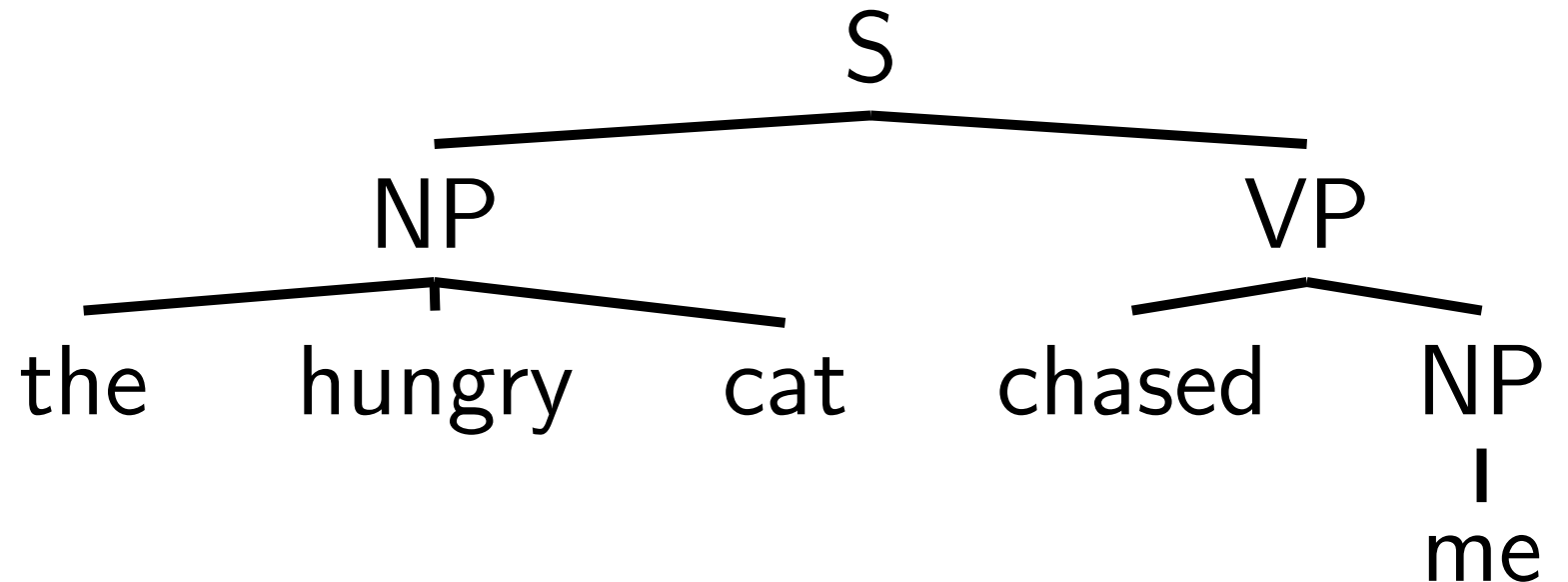
REDUCE

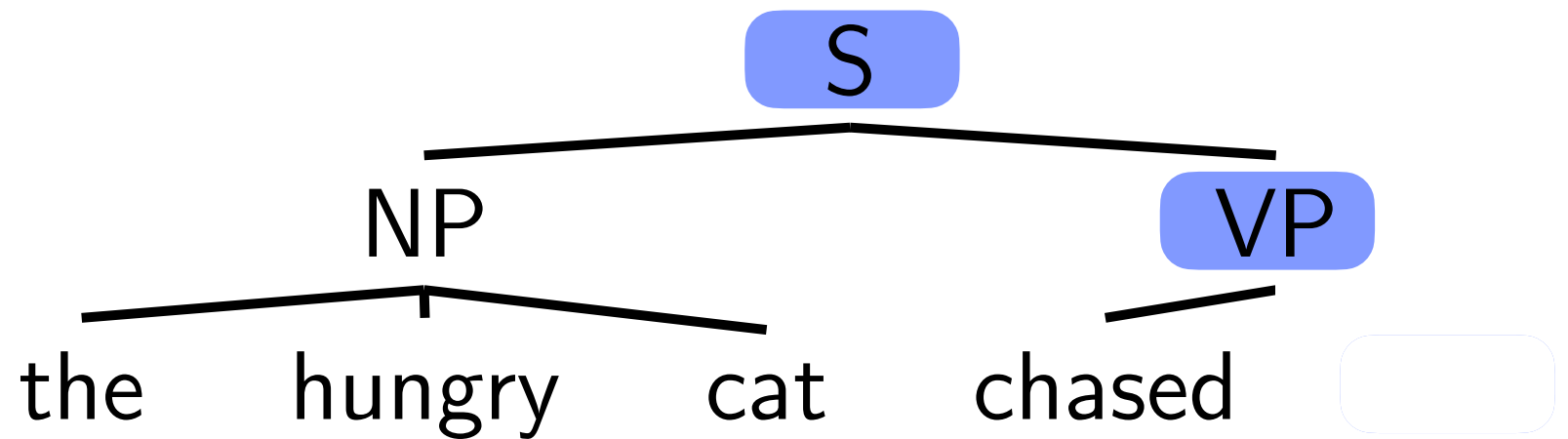
(S | (NP the hungry cat) | (VP chased (NP me))

REDUCE

(S (NP the hungry cat) (VP chased (NP me)))

END





Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

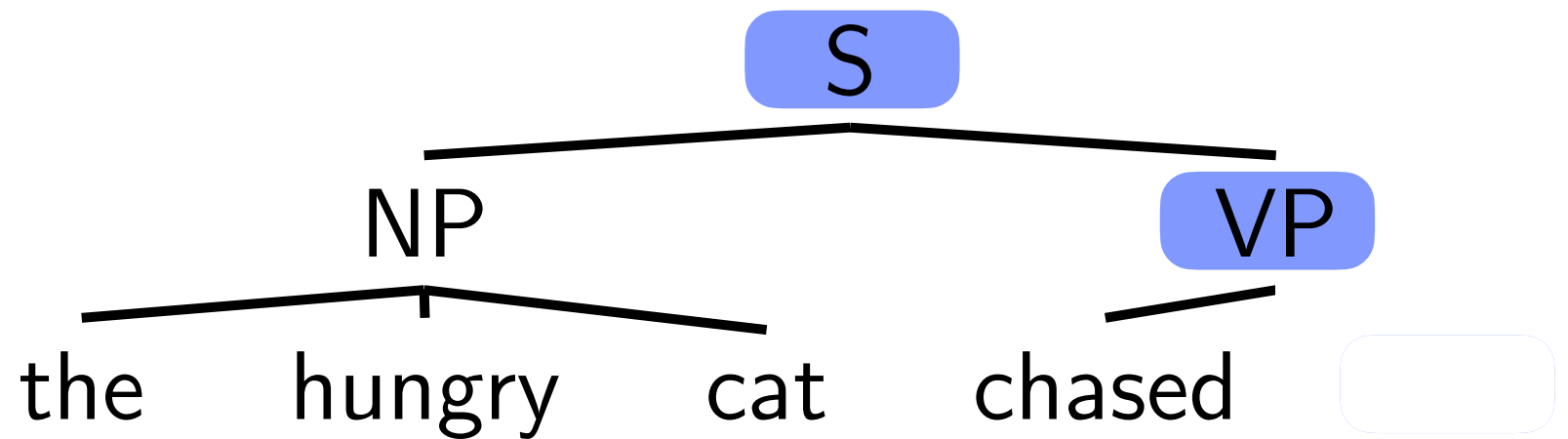
NT(VP)

(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

???



Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

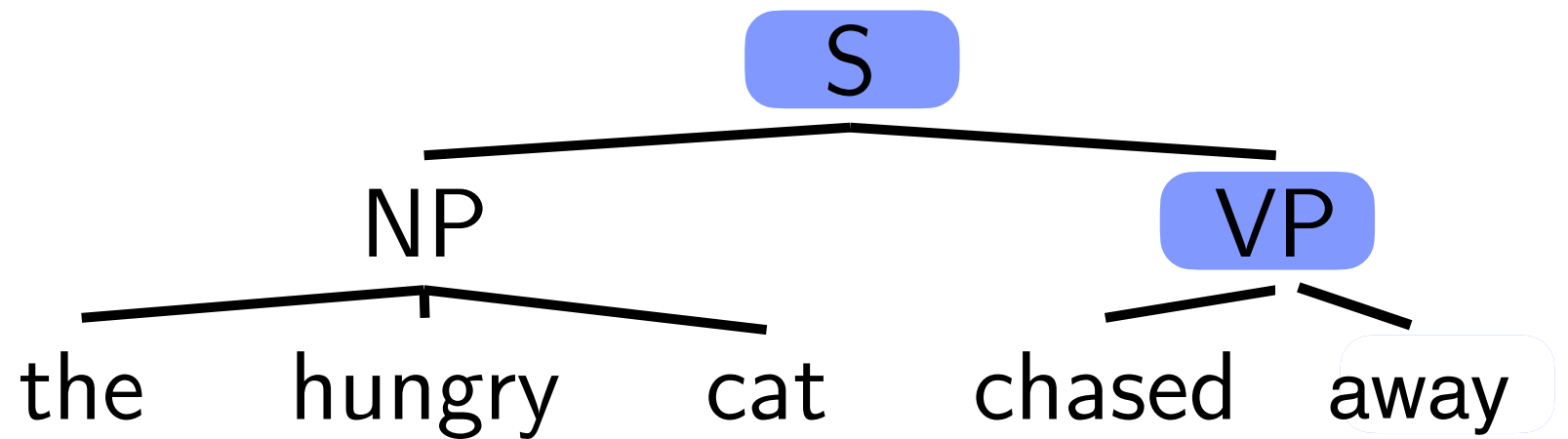
(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

???

Gen(away)



Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

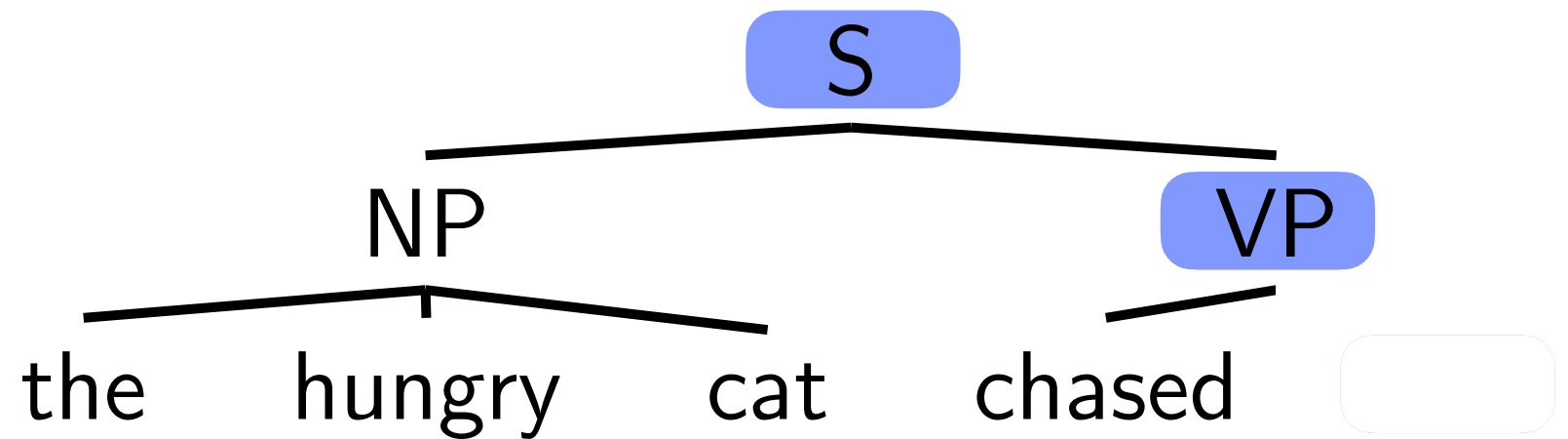
(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

???

Gen(away)



Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

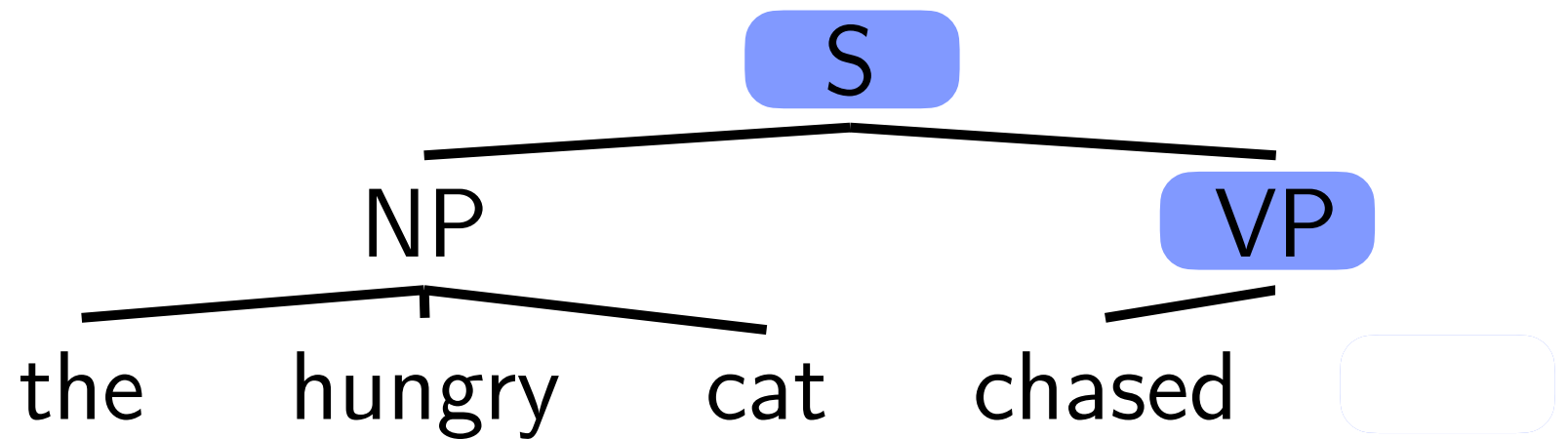
(S | (NP the hungry cat) | (VP

Gen(chased)

(S | (NP the hungry cat) | (VP | chased

???

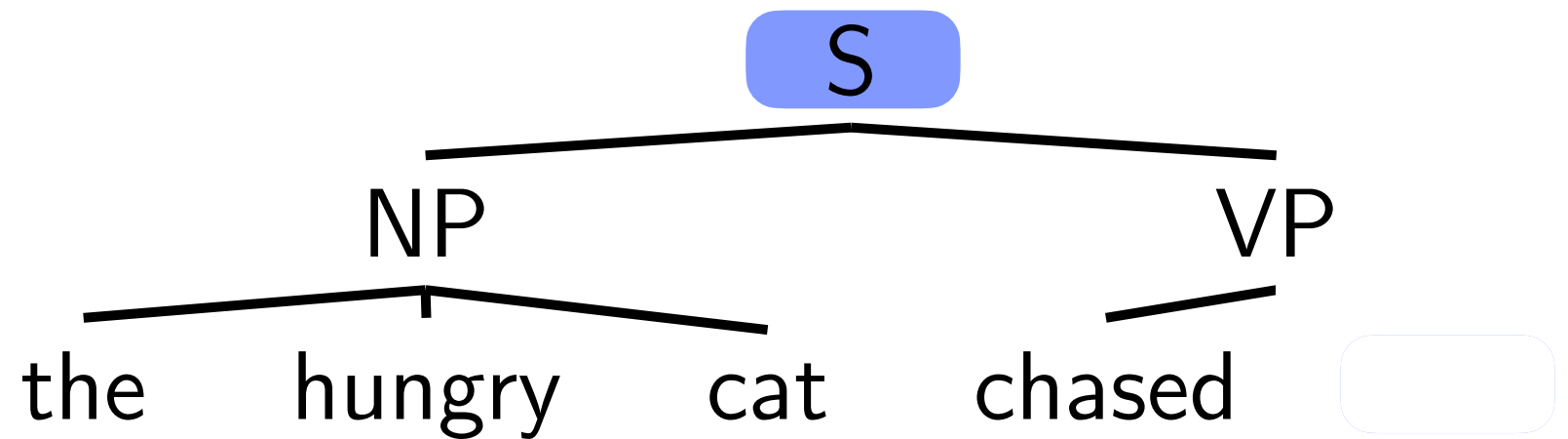
Gen(away)



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

Gen(away) REDUCE



Action

Stack

NT(S)

(S

NT(NP)

(S | (NP

Gen(the)

(S | (NP | the

Gen(hungry)

(S | (NP | the | hungry

Gen(cat)

(S | (NP | the | hungry | cat

REDUCE

(S | (NP the hungry cat)

NT(VP)

(S | (NP the hungry cat) | (VP

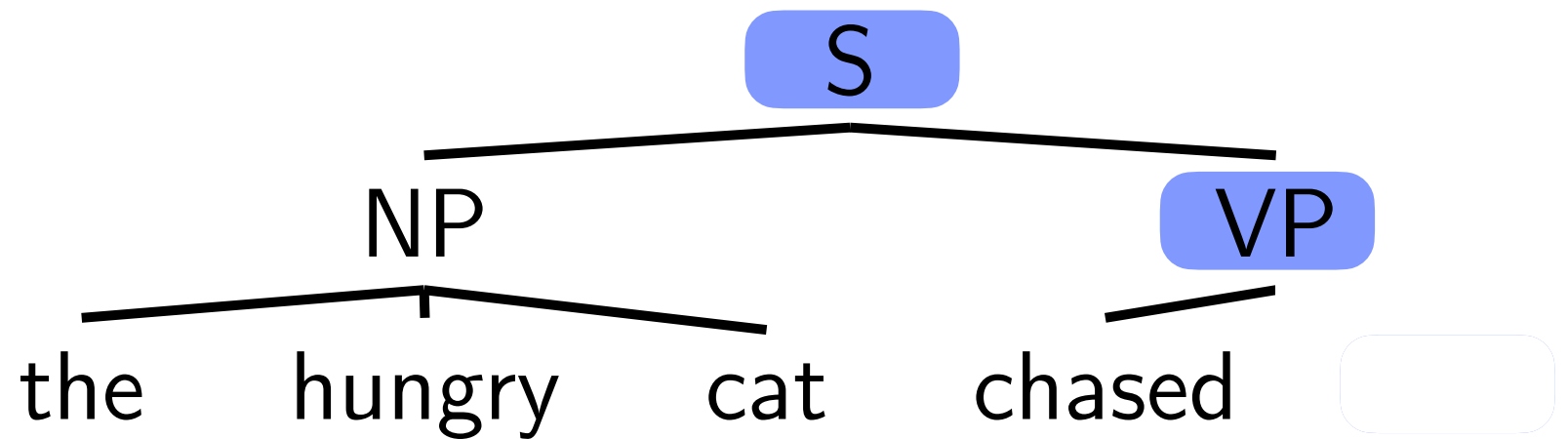
Gen(chased)

(S | (NP the hungry cat) | (VP | chased

???

Gen(away)

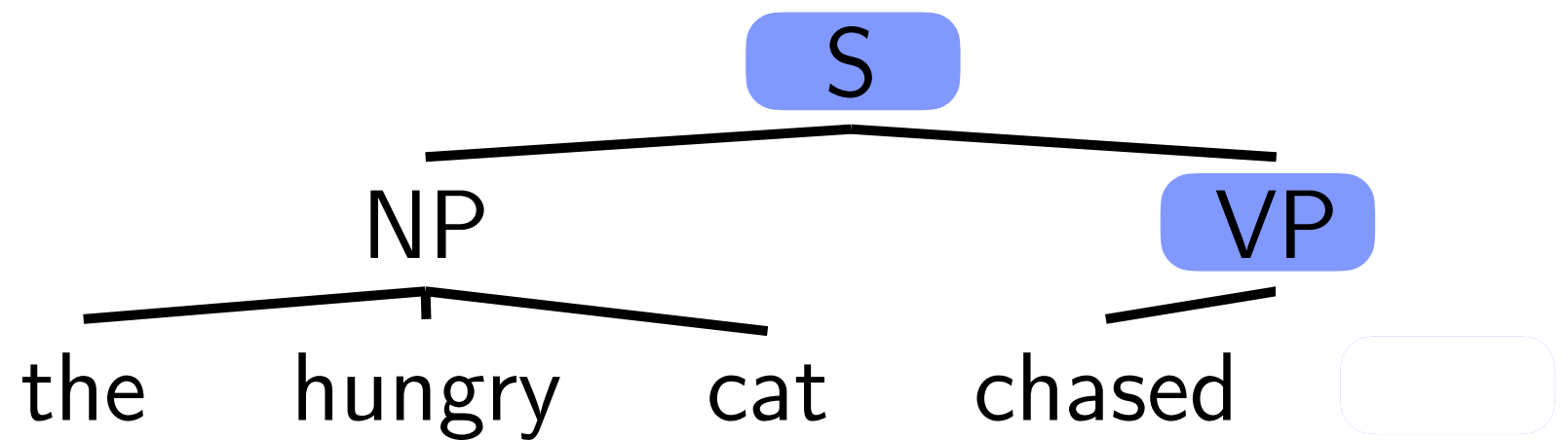
REDUCE



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

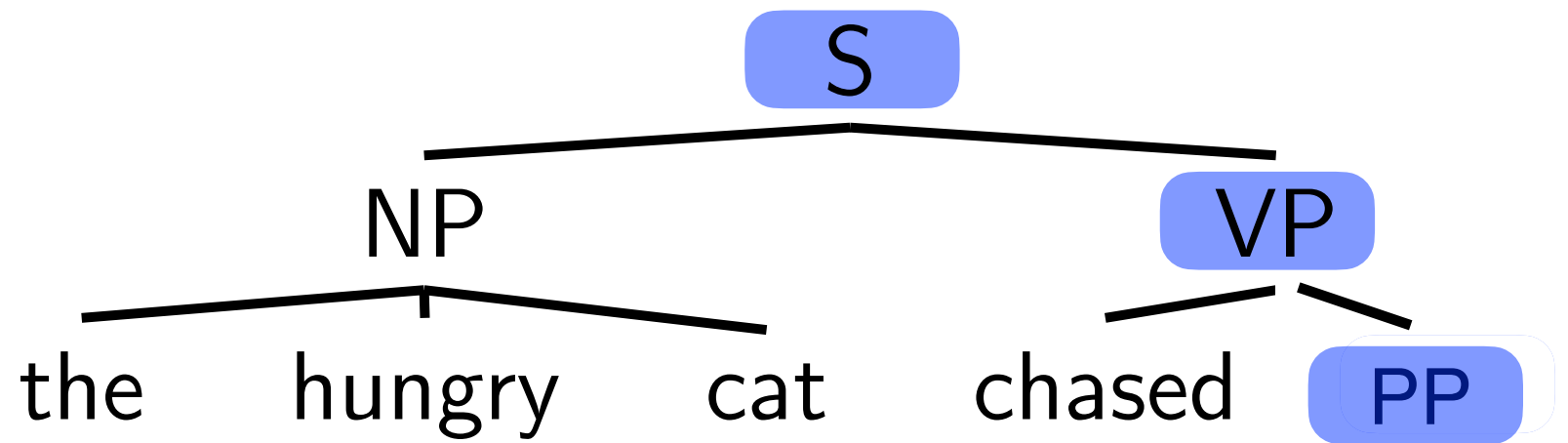
Gen(away) REDUCE



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

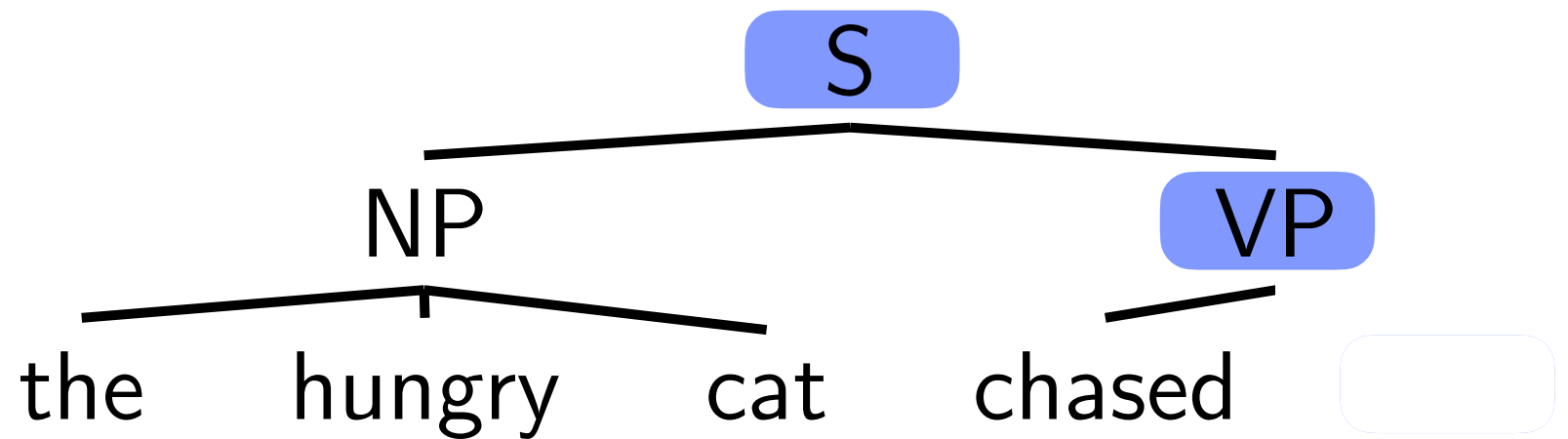
Gen(away) REDUCE NT(PP)



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

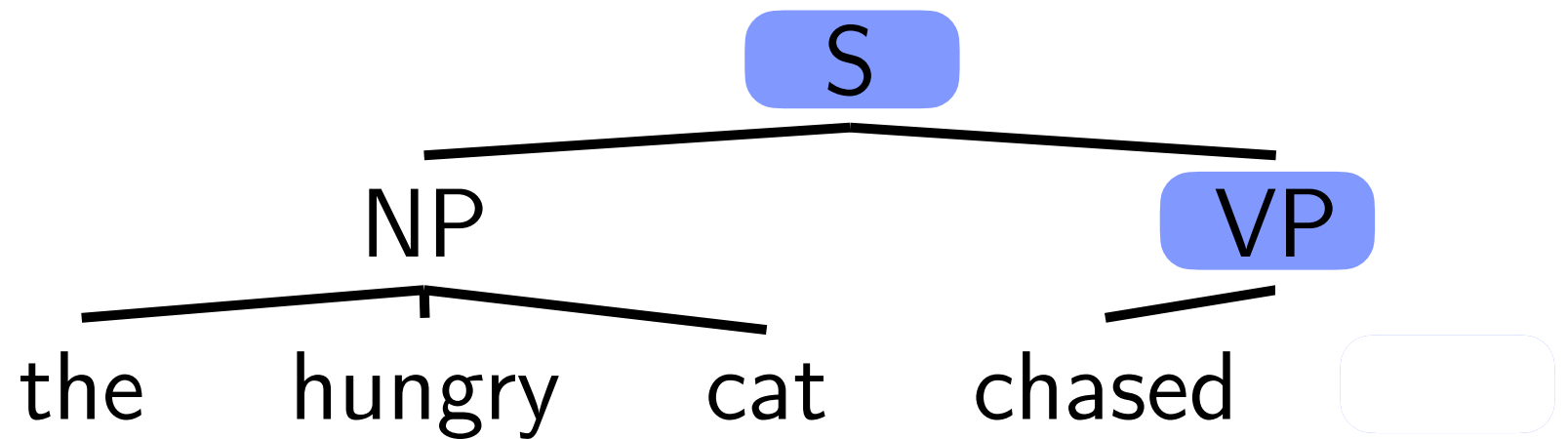
Gen(away) REDUCE NT(PP)



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

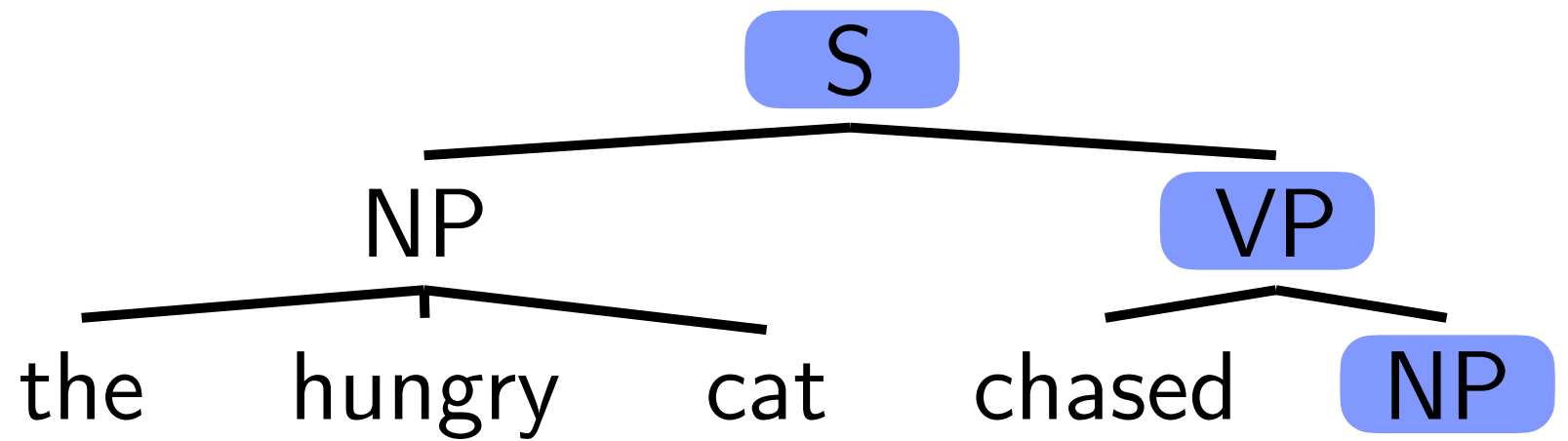
Gen(away) REDUCE NT(PP)



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

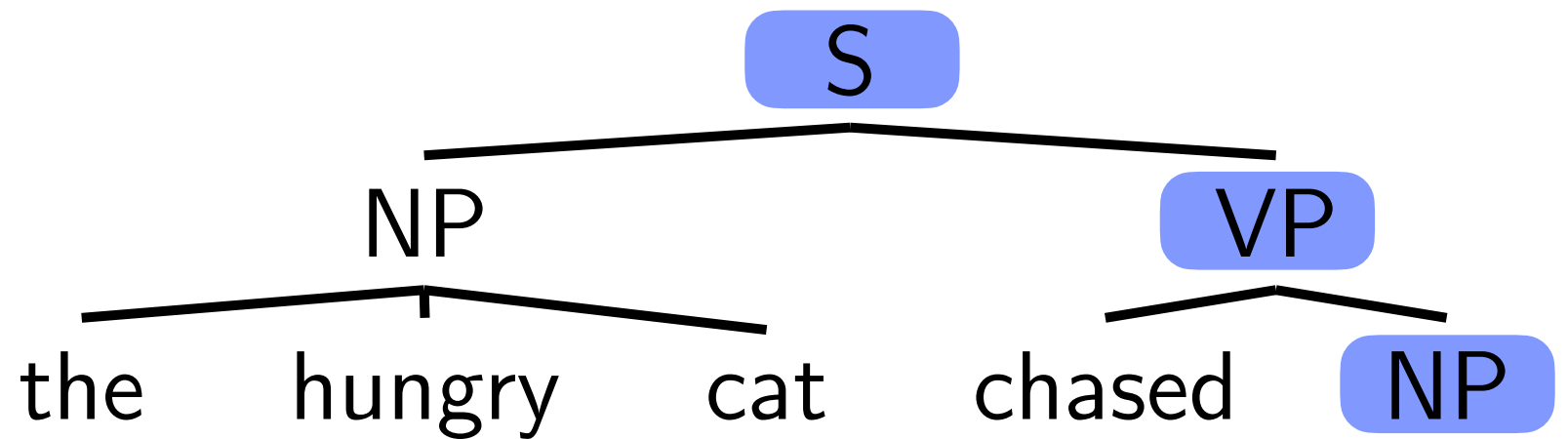
Gen(away) REDUCE NT(PP) NT(NP)



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

Gen(away) REDUCE NT(PP) NT(NP)

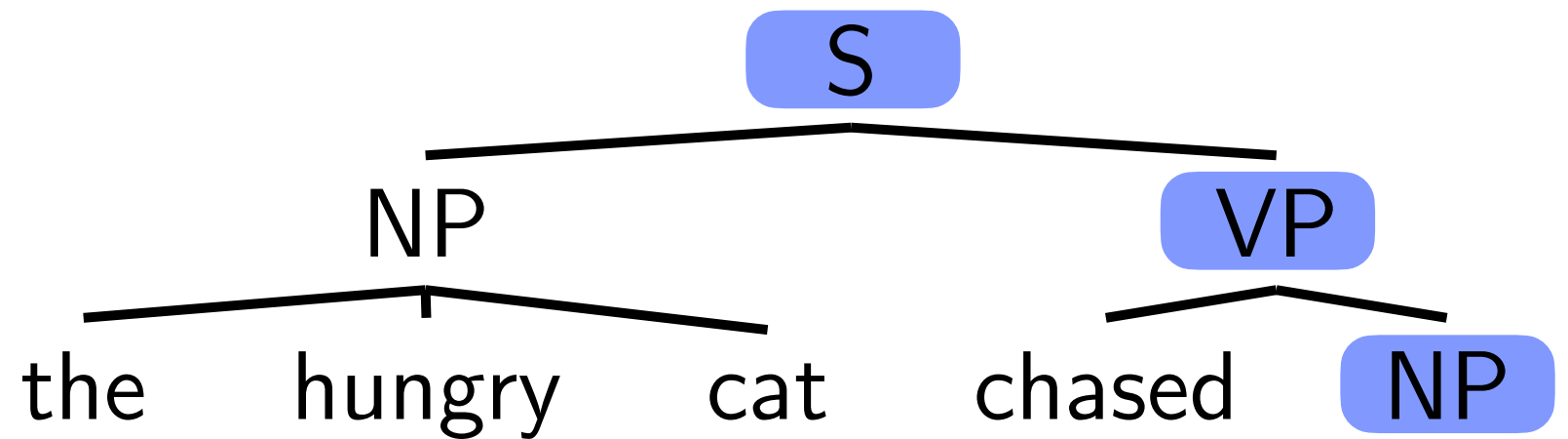


Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

Gen(away) REDUCE NT(PP) NT(NP)

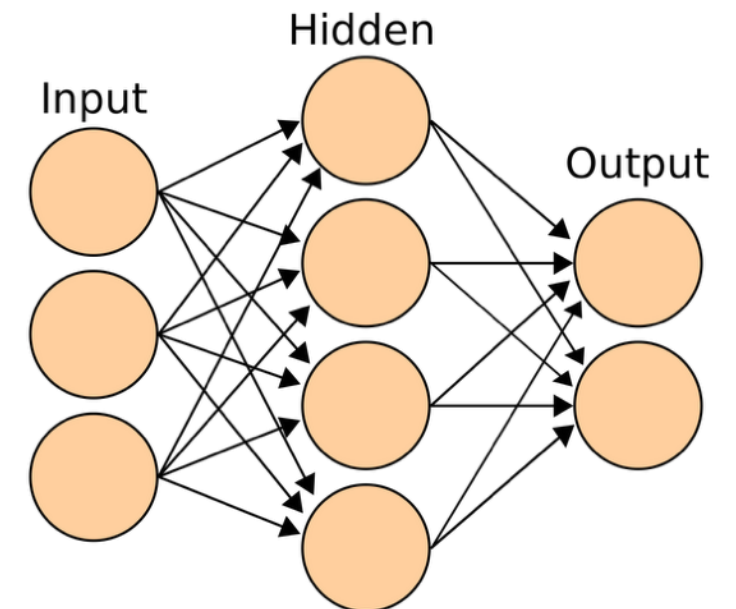
Knowledge characterization: P(action|context)



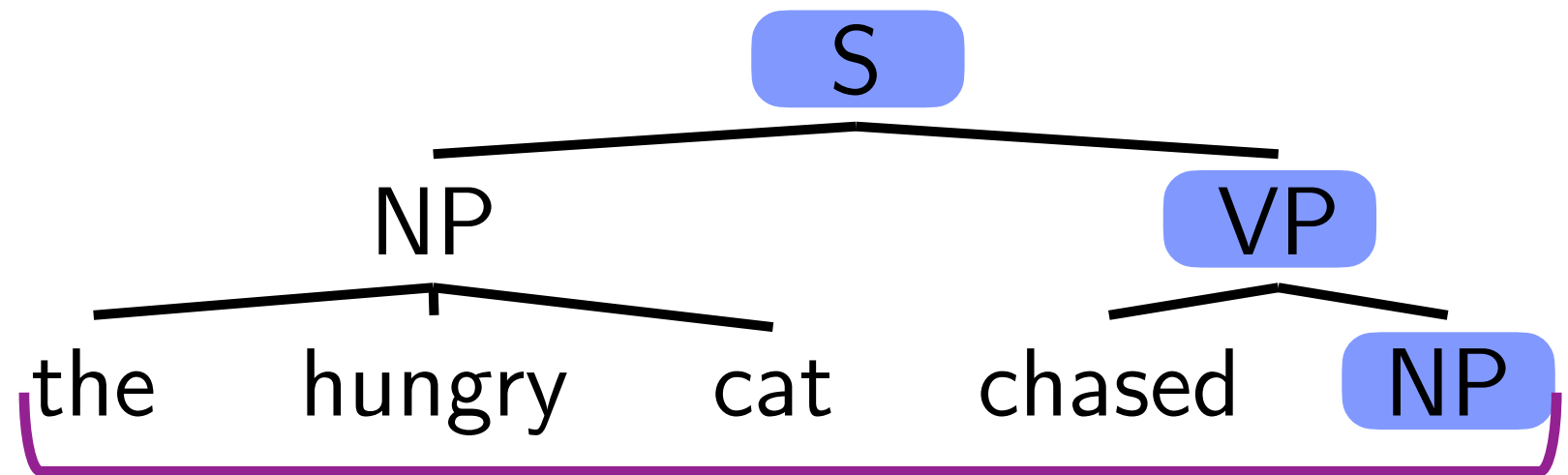
Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

Gen(away) REDUCE NT(PP) NT(NP)



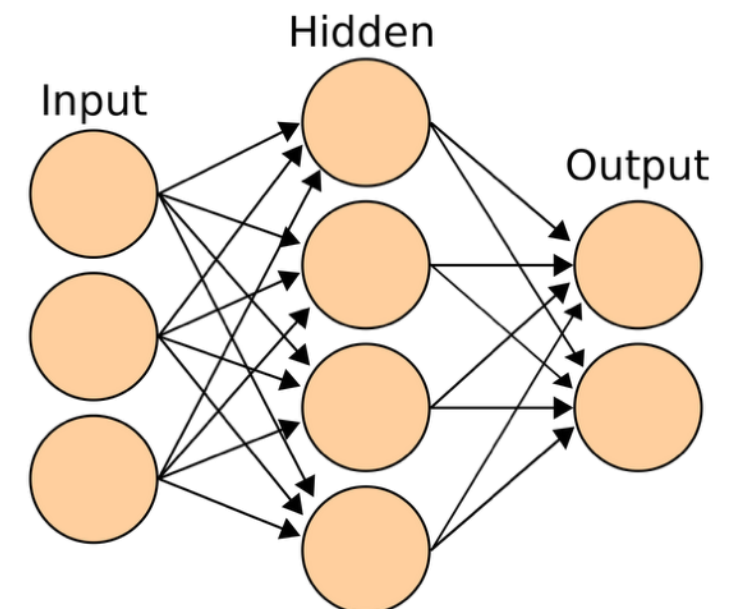
Knowledge characterization: $P(\text{action}|\text{context})$



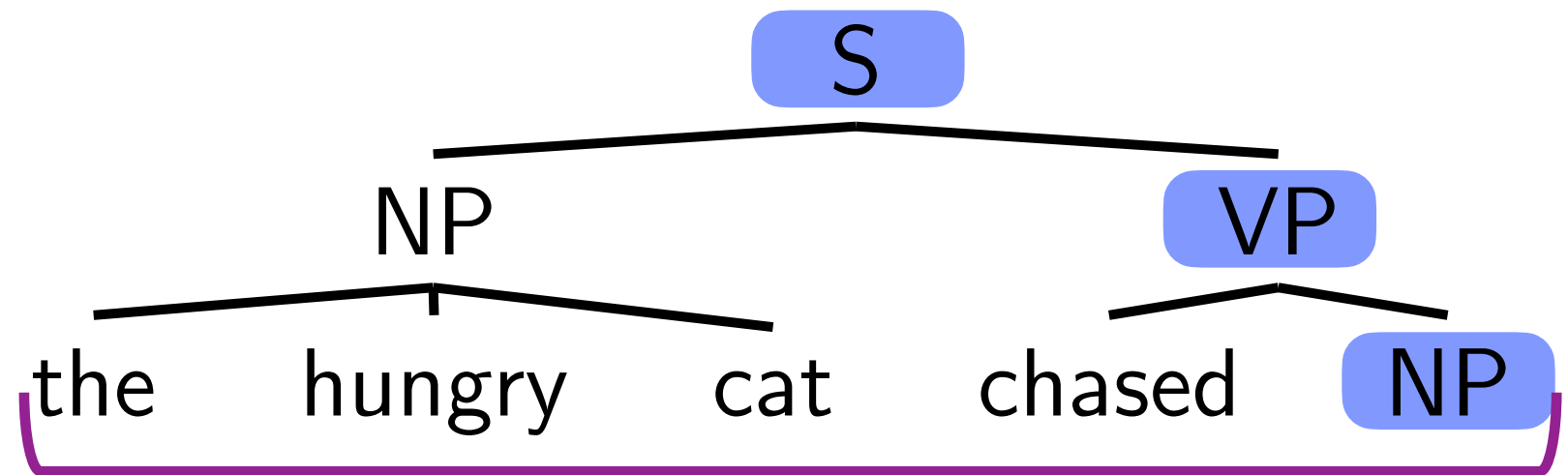
Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

???

Gen(away) REDUCE NT(PP) NT(NP)



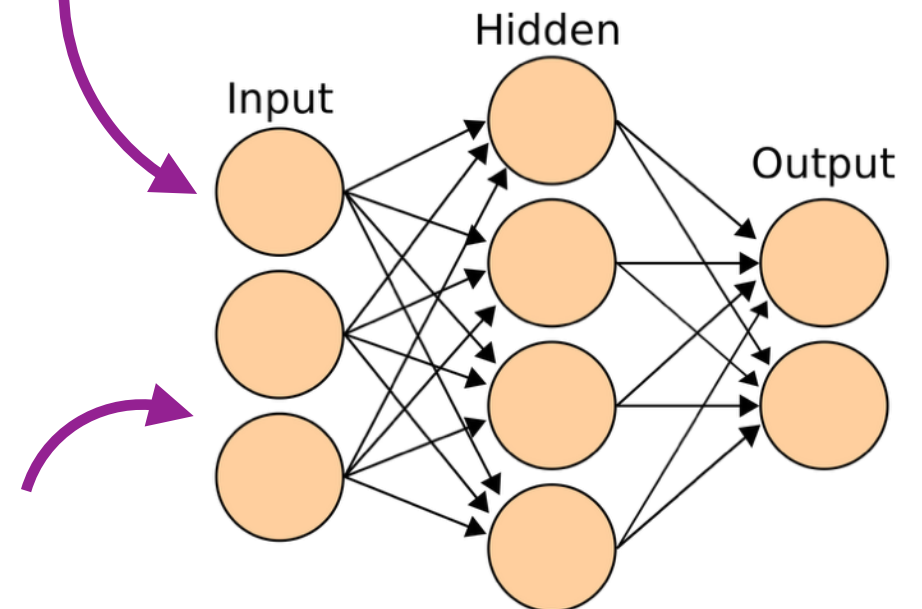
Knowledge characterization: $P(\text{action}|\text{context})$



Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased

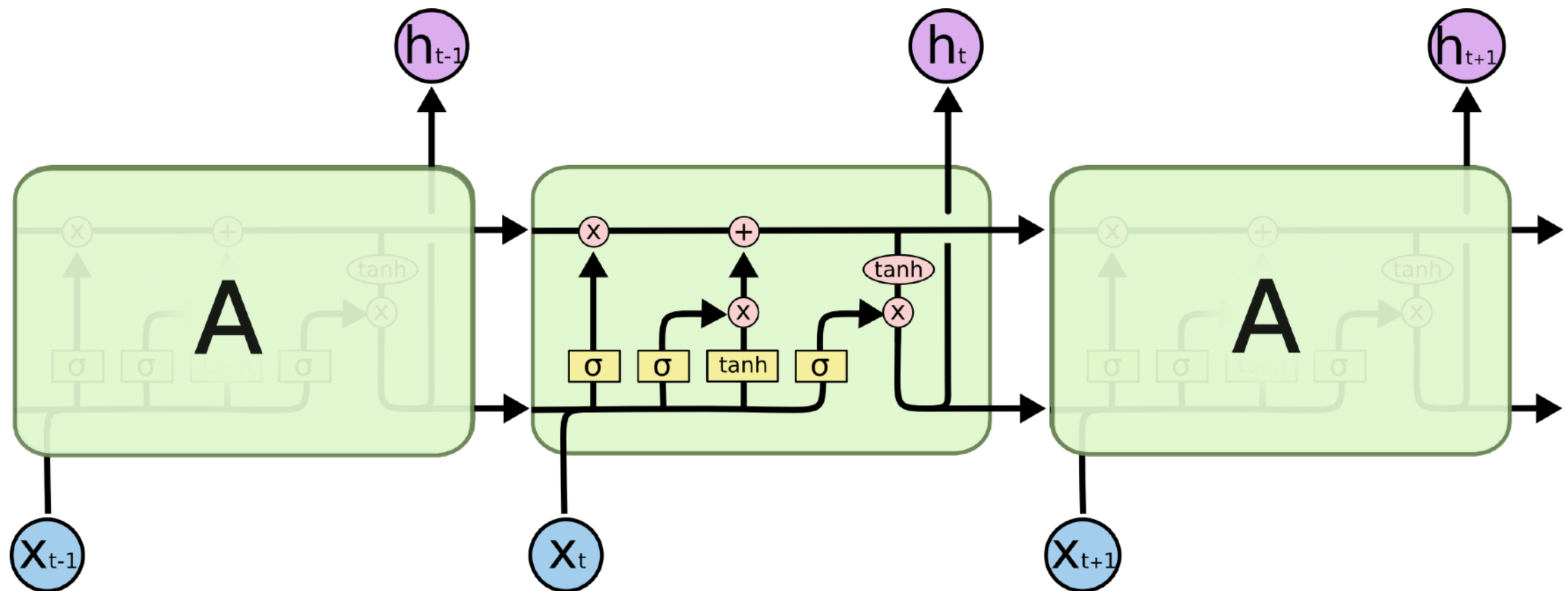
???

Gen(away) REDUCE NT(PP) NT(NP)



Knowledge characterization: $P(\text{action}|\text{context})$

Our friend the LSTM



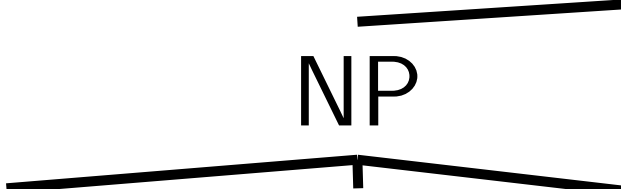
Some options for neural generalization

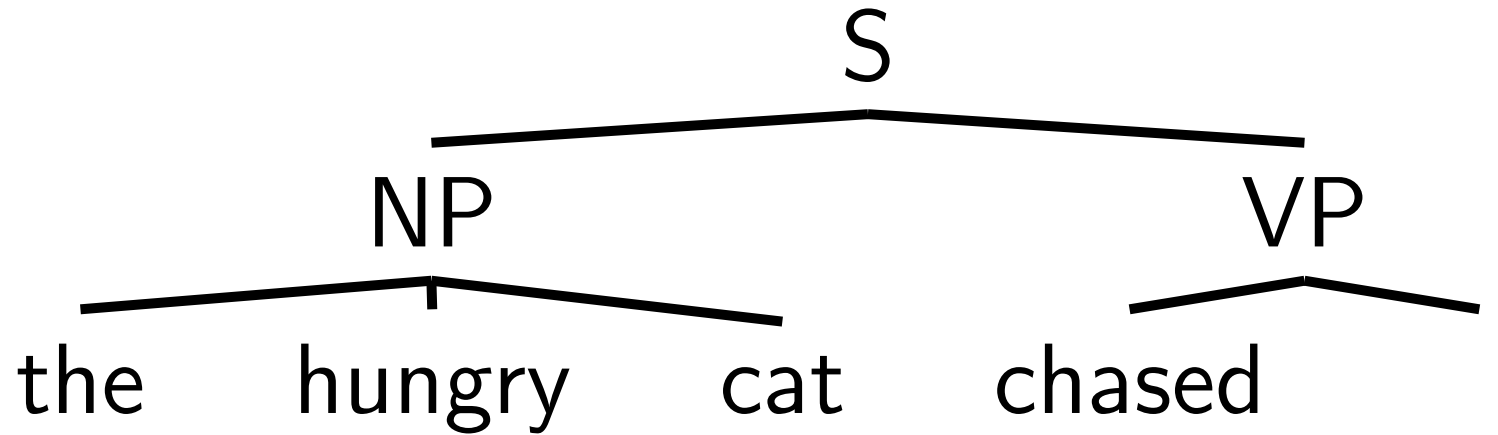
- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions

						S		
Action	Stack							
NT(S)	(S					NP		VP
NT(NP)	(S (NP	the	hungry	cat			chased	
Gen(the)	(S (NP the							
Gen(hungry)	(S (NP the hungry							
Gen(cat)	(S (NP the hungry cat							
REDUCE	(S (NP the hungry cat)							
NT(VP)	(S (NP the hungry cat) (VP							
Gen(chased)	(S (NP the hungry cat) (VP chased							

Some options for neural generalization

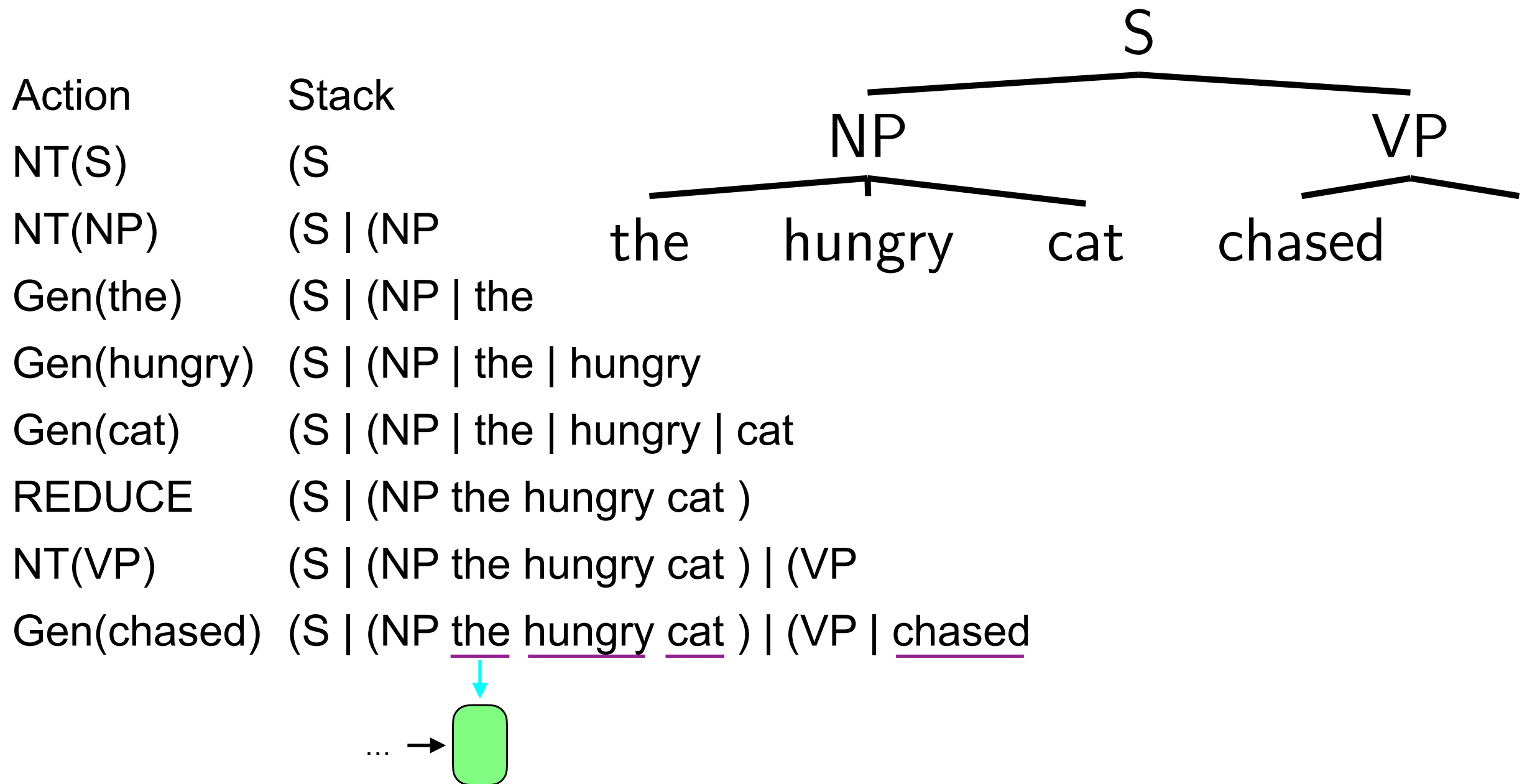
- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions

Action	Stack	
NT(S)	(S	
NT(NP)	(S (NP	the hungry
Gen(the)	(S (NP the	
Gen(hungry)	(S (NP the hungry	
Gen(cat)	(S (NP the hungry cat	
REDUCE	(S (NP the hungry cat)	
NT(VP)	(S (NP the hungry cat) (VP	
Gen(chased)	(S (NP the hungry cat) (VP chased	



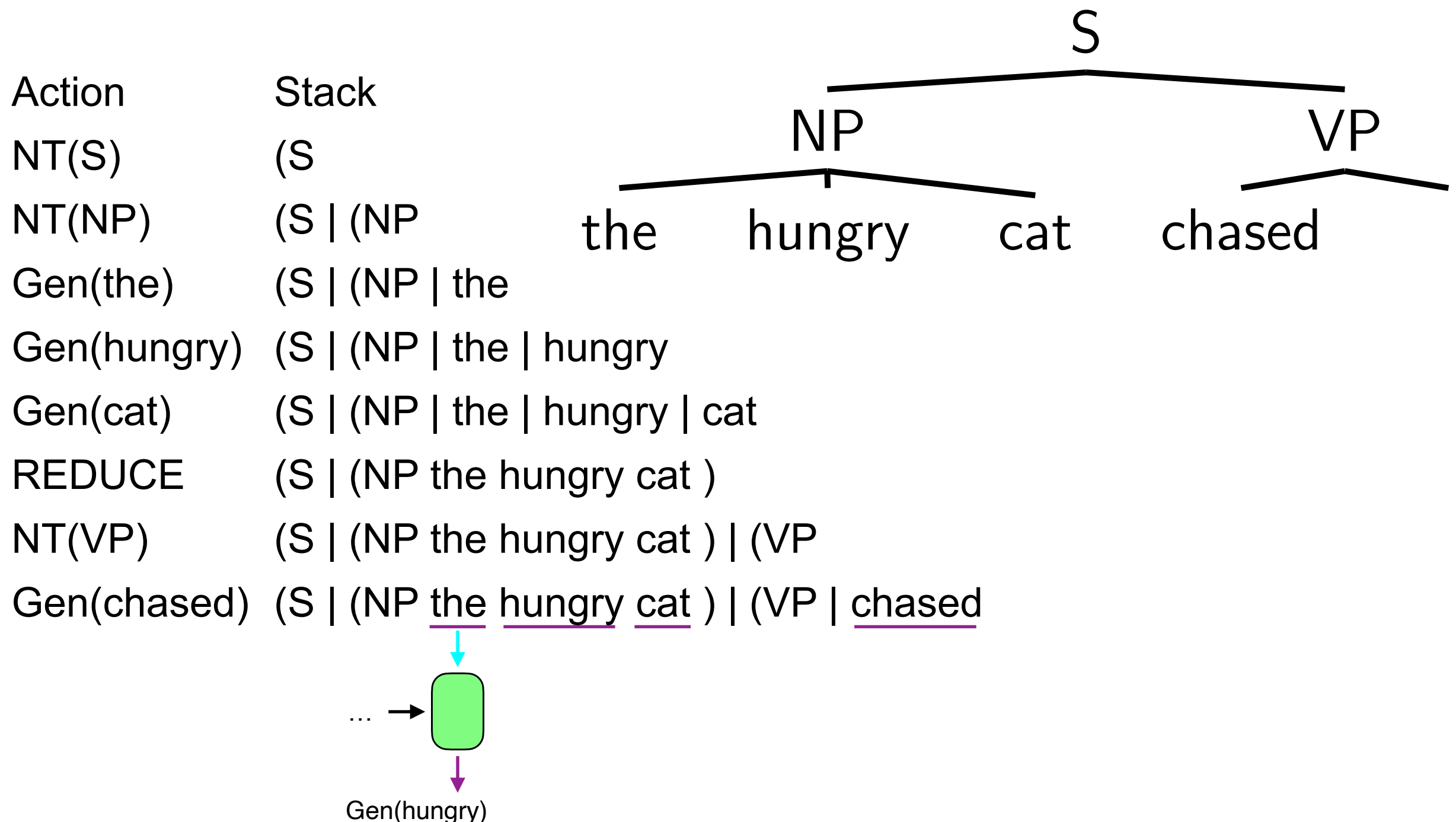
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



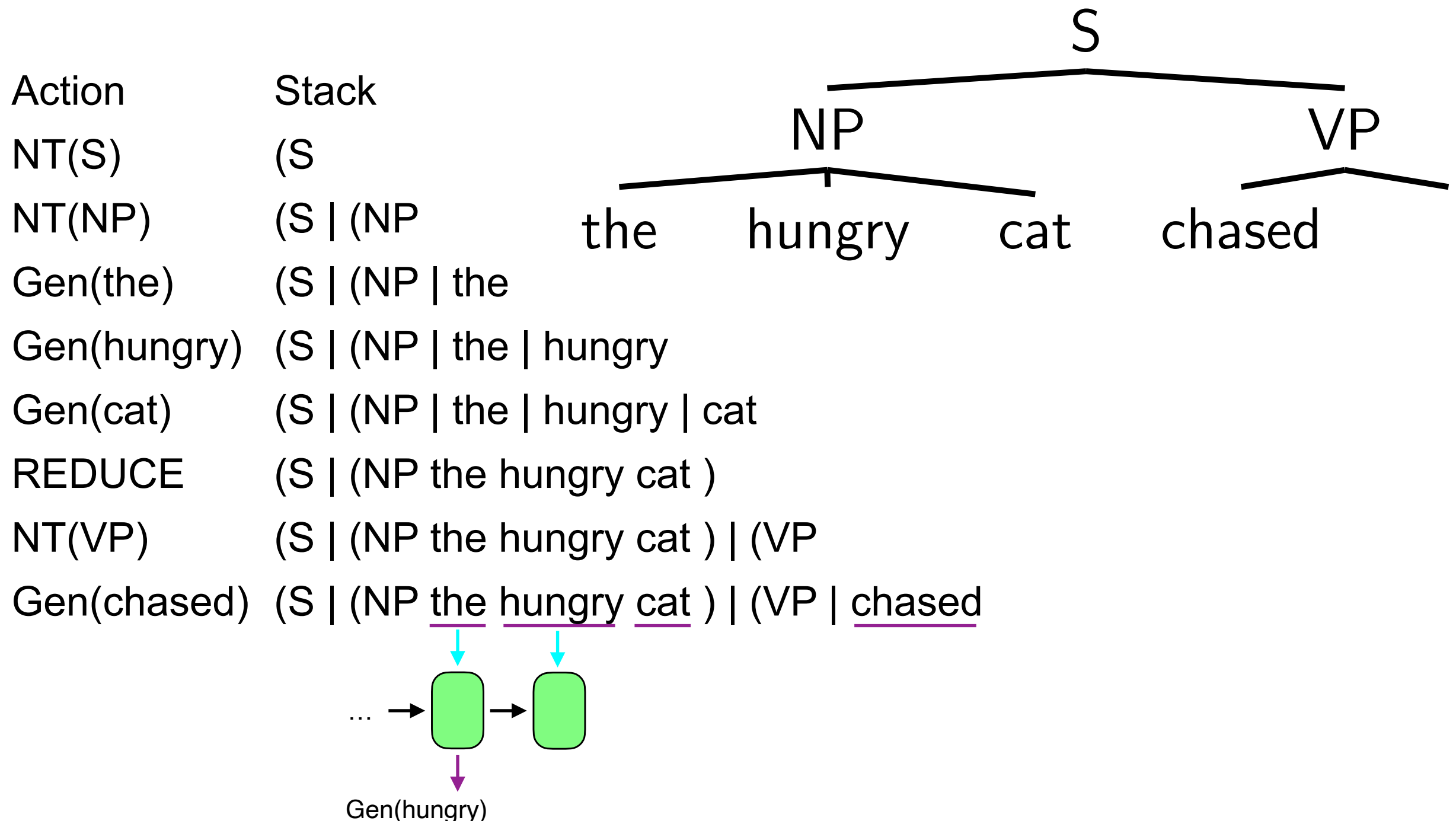
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



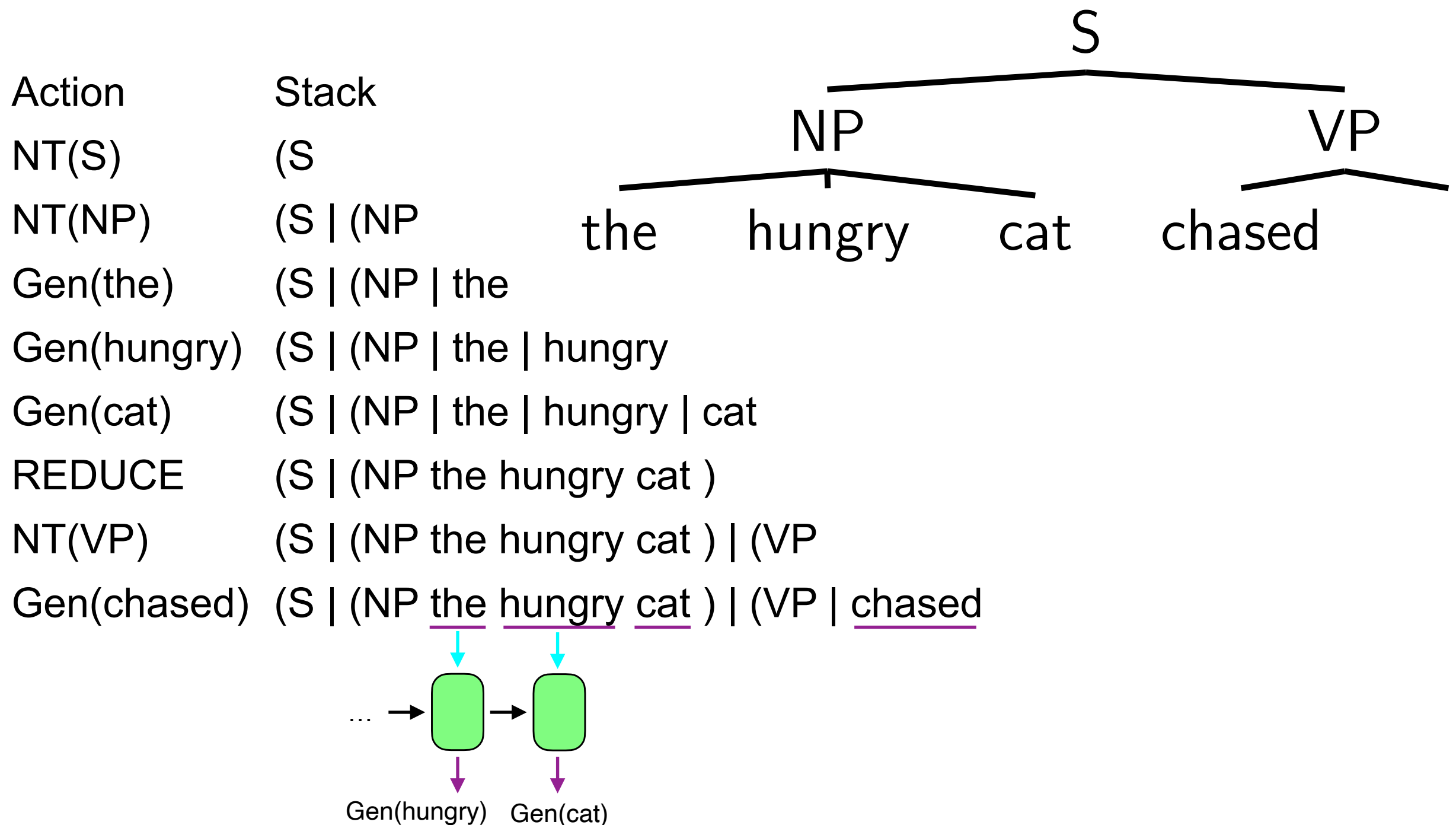
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



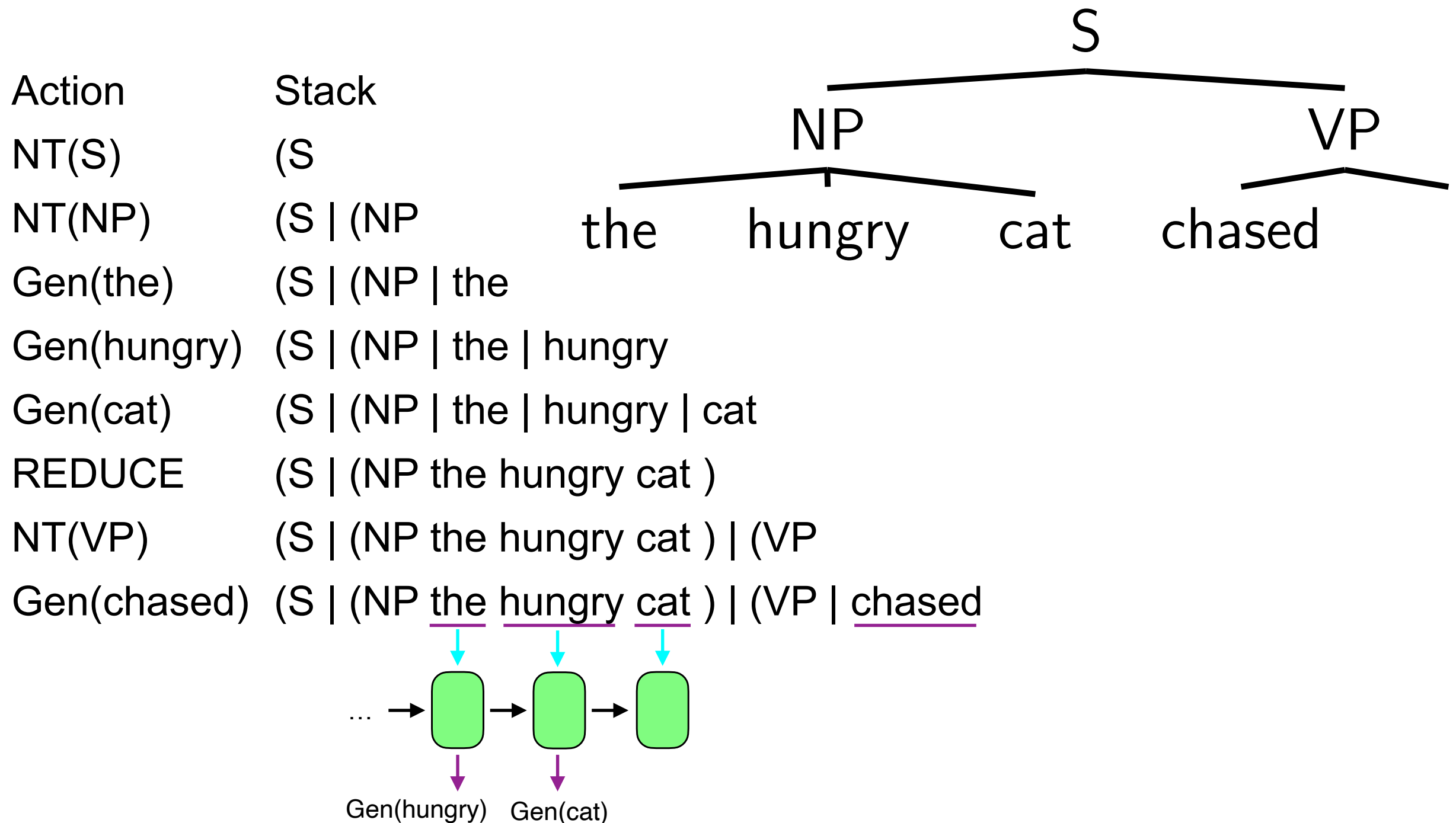
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



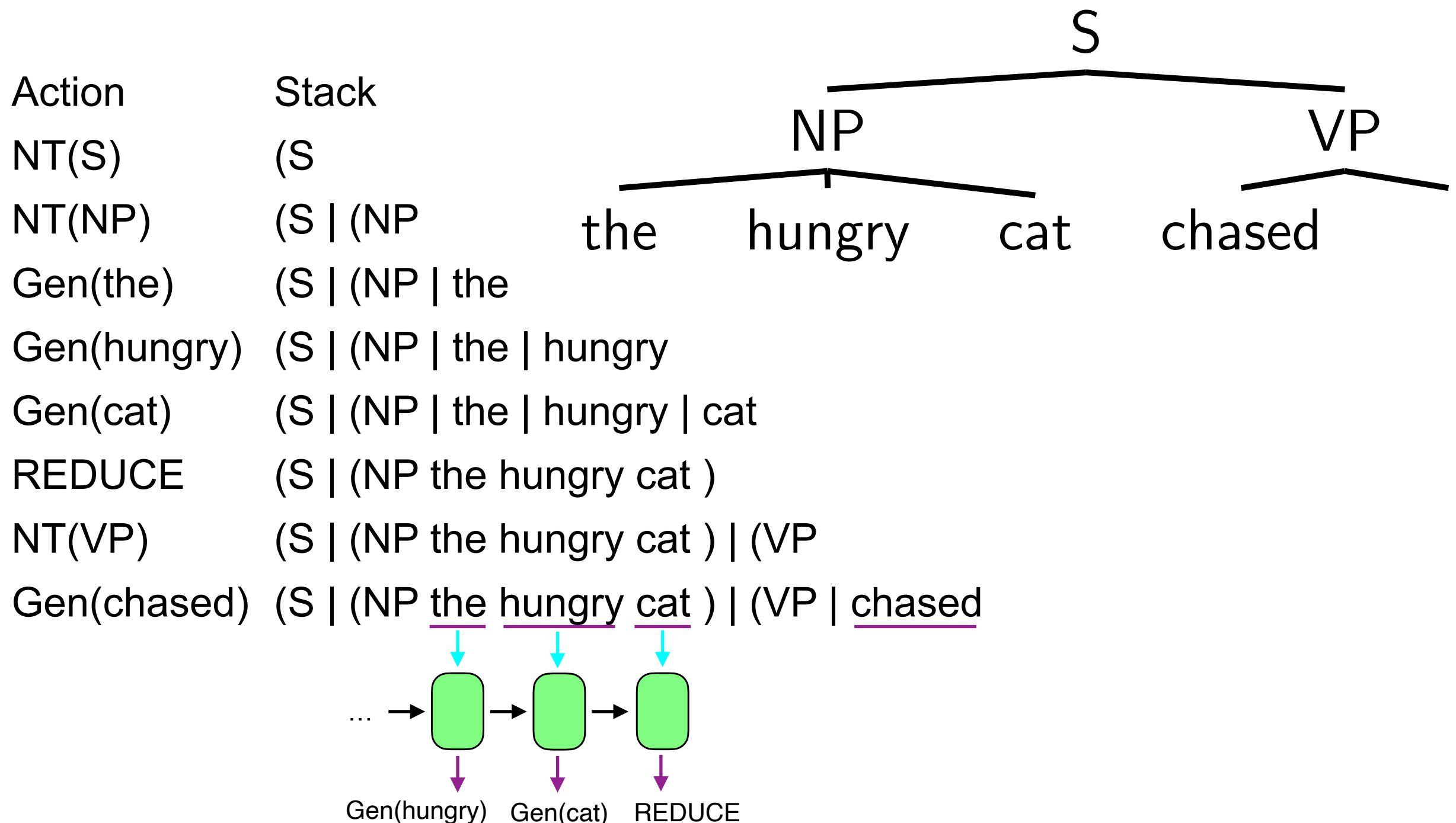
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



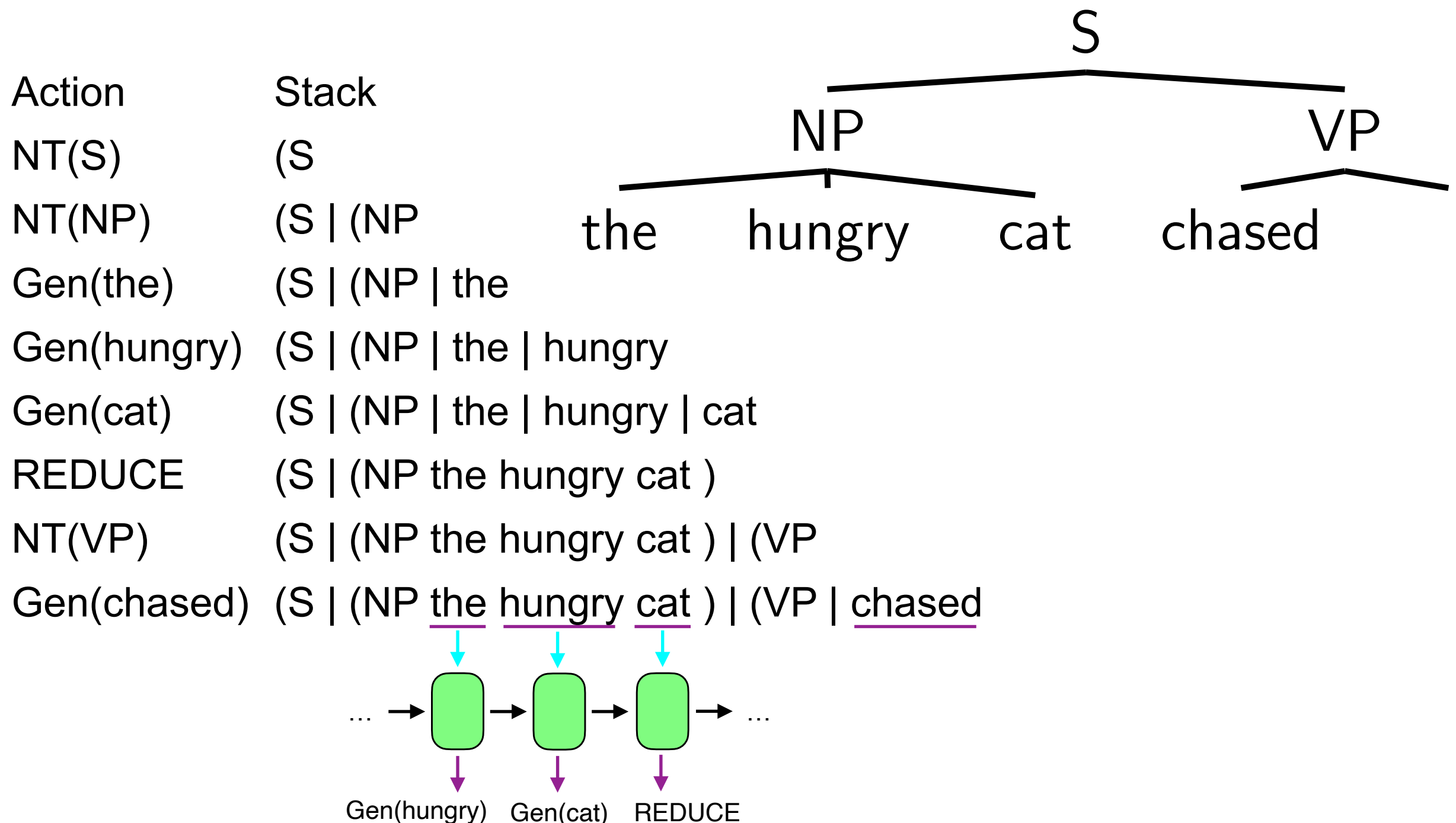
Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



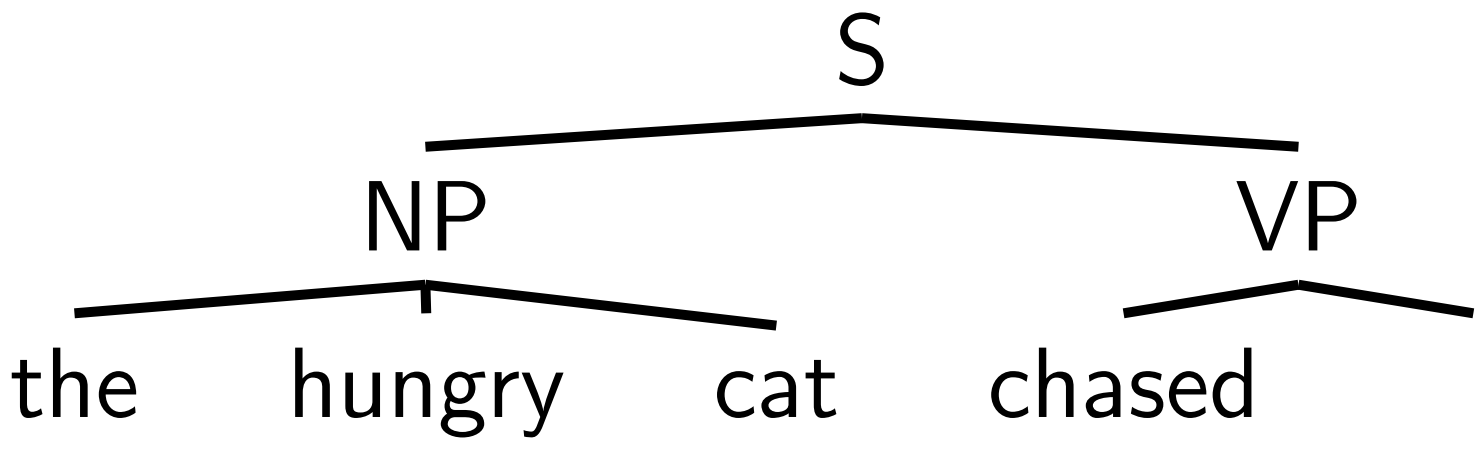
Some options for neural generalization

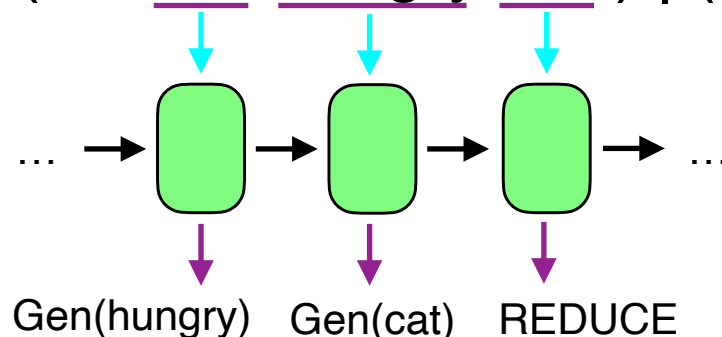
- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions



Some options for neural generalization

- Option 1: run RNN over words as normal for language modeling, but predict tree-generation actions

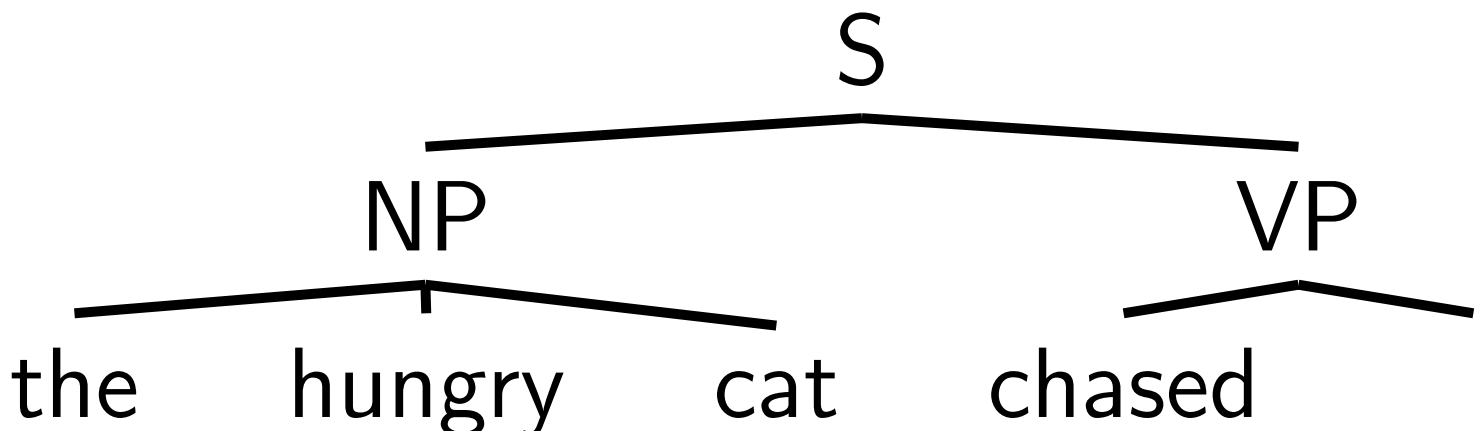
Action	Stack	
NT(S)	(S	
NT(NP)	(S (NP	
Gen(the)	(S (NP the	
Gen(hungry)	(S (NP the hungry	
Gen(cat)	(S (NP the hungry cat	
REDUCE	(S (NP the hungry cat)	
NT(VP)	(S (NP the hungry cat) (VP	
Gen(chased)	(S (NP <u>the hungry cat</u>) (VP <u>chased</u>	



This will not work well at all—why???

Some options for neural generalization

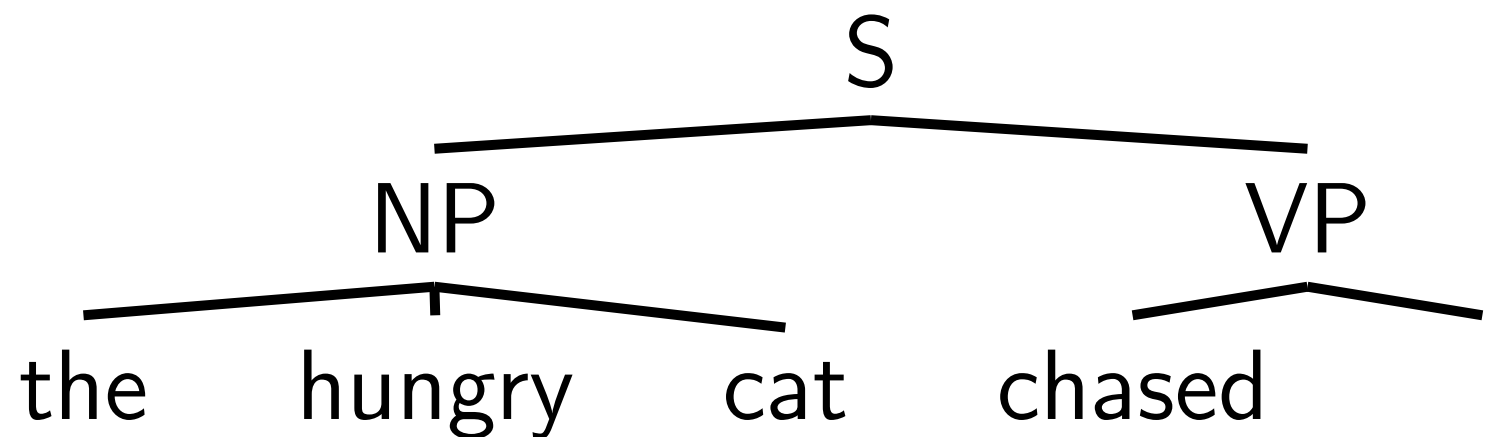
- Option 2: run RNN over *tree-generation actions*

Action	Stack				
NT(S)	(S				
NT(NP)	(S (NP	the	hungry	cat	chased
Gen(the)	(S (NP the				
Gen(hungry)	(S (NP the hungry				
Gen(cat)	(S (NP the hungry cat				
REDUCE	(S (NP the hungry cat)				
NT(VP)	(S (NP the hungry cat) (VP				
Gen(chased)	(S (NP the hungry cat) (VP chased				

Some options for neural generalization

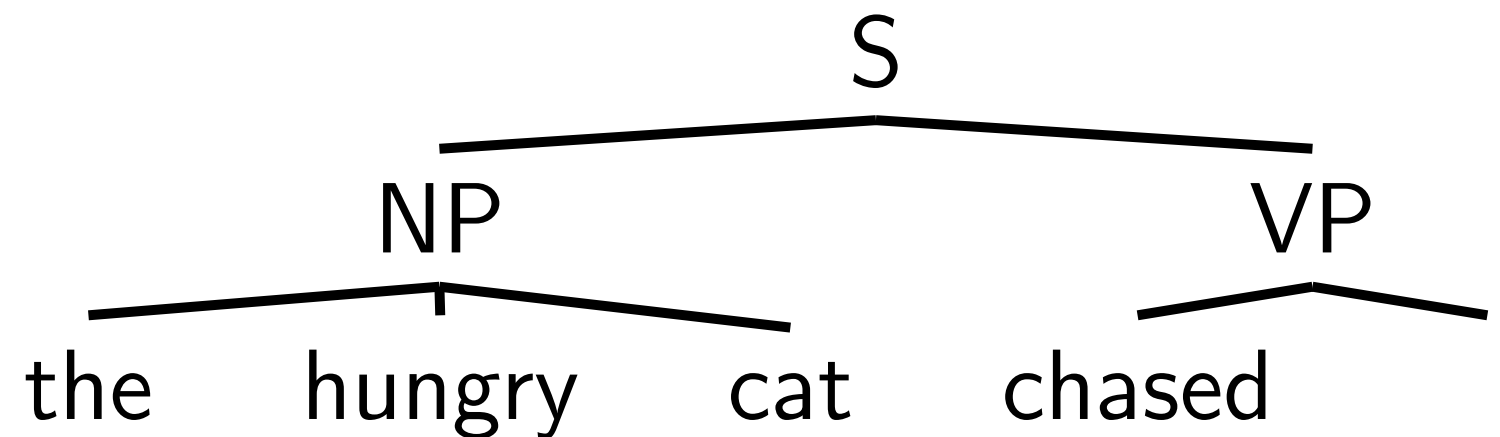
- Option 2: run RNN over *tree-generation actions*

Action	Stack
NT(S)	(S
NT(NP)	(S (NP
Gen(the)	(S (NP the
Gen(hungry)	(S (NP the hungry
Gen(cat)	(S (NP the hungry cat
REDUCE	(S (NP the hungry cat)
NT(VP)	(S (NP the hungry cat) (VP
Gen(chased)	(S (NP the hungry cat) (VP chased



Some options for neural generalization

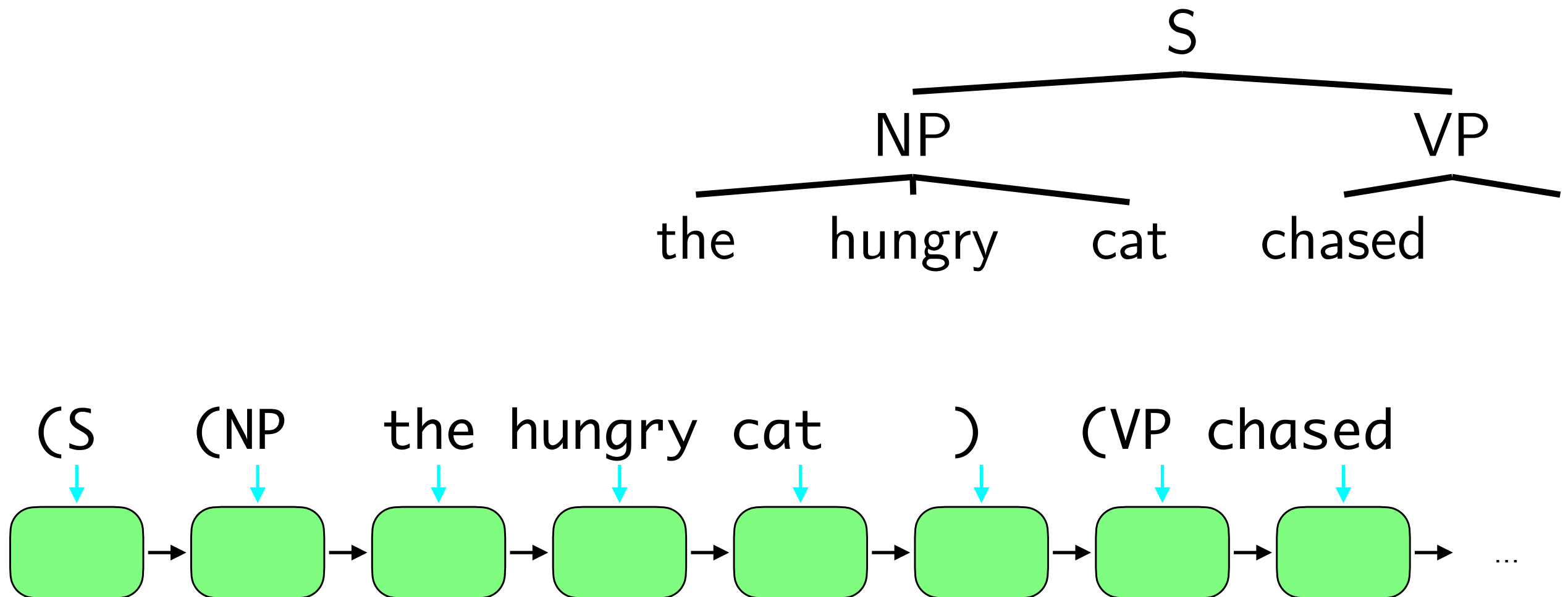
- Option 2: run RNN over *tree generation actions*



(S (NP the hungry cat) (VP chased

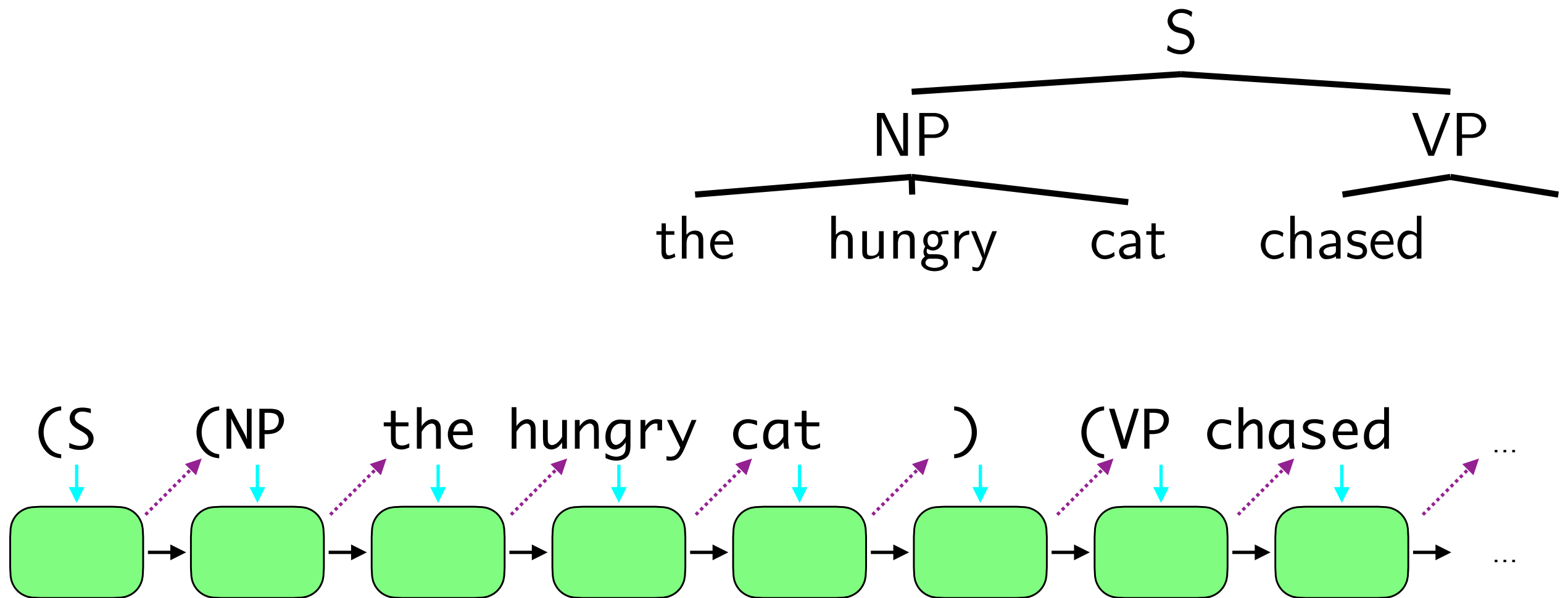
Some options for neural generalization

- Option 2: run RNN over *tree generation actions*



Some options for neural generalization

- Option 2: run RNN over *tree generation actions*



An inferential challenge

(S (NP I) (VP saw the

An inferential challenge

(S (NP I) (VP saw (NP the

An inferential challenge

(S (NP I) (VP saw (NP the

I saw the child

An inferential challenge

(S (NP I) (VP saw (NP the

I saw the child

(S (NP I) (VP saw (NP (NP the

I saw the child's dog

An inferential challenge

(S (NP I) (VP saw (NP the

I saw the child

(S (NP I) (VP saw (NP (NP the

I saw the child's dog

(S (NP I) (VP saw (S (NP the

I saw the child leave

An inferential challenge

(S (NP I) (VP saw (NP the	<i>I saw the child</i>
(S (NP I) (VP saw (NP (NP the	<i>I saw the child's dog</i>
(S (NP I) (VP saw (S (NP the	<i>I saw the child leave</i>
(S (NP I) (VP saw (S (NP (NP the	<i>I saw the child's dog leave</i>

An inferential challenge

(S (NP I) (VP saw (NP the	<i>I saw the child</i>
(S (NP I) (VP saw (NP (NP the	<i>I saw the child's dog</i>
(S (NP I) (VP saw (S (NP the	<i>I saw the child leave</i>
(S (NP I) (VP saw (S (NP (NP the	<i>I saw the child's dog leave</i>
(S (NP I) (VP saw (SBAR (NP the	<i>I saw the child left</i>

An inferential challenge

(S (NP I) (VP saw (NP the	<i>I saw the child</i>
(S (NP I) (VP saw (NP (NP the	<i>I saw the child's dog</i>
(S (NP I) (VP saw (S (NP the	<i>I saw the child leave</i>
(S (NP I) (VP saw (S (NP (NP the	<i>I saw the child's dog leave</i>
(S (NP I) (VP saw (SBAR (NP the	<i>I saw the child left</i>
(S (NP I) (VP saw (SBAR (NP (NP the	<i>I saw the child's dog left</i>

An inferential challenge

(S (NP I) (VP saw (NP the	<i>I saw the child</i>
(S (NP I) (VP saw (NP (NP the	<i>I saw the child's dog</i>
(S (NP I) (VP saw (S (NP the	<i>I saw the child leave</i>
(S (NP I) (VP saw (S (NP (NP the	<i>I saw the child's dog leave</i>
(S (NP I) (VP saw (SBAR (NP the	<i>I saw the child left</i>
(S (NP I) (VP saw (SBAR (NP (NP the	<i>I saw the child's dog left</i>

There is a potentially unbounded number of tree-generation operations just to get to the next word!

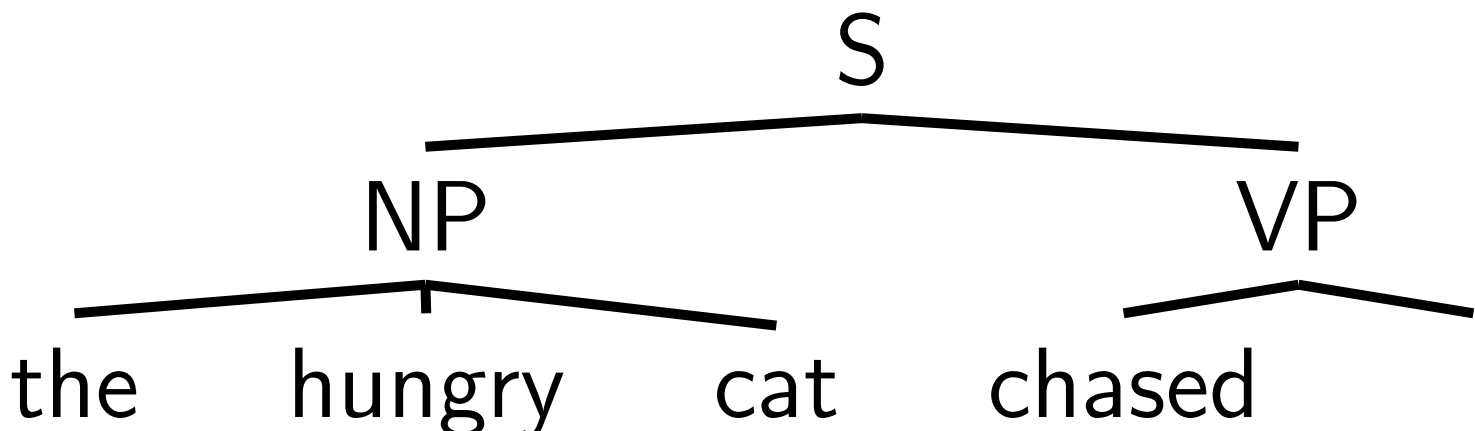
Inference using beam search

Context C	Action Sequences A	$\log P(A C)$	Rank on beam
(S (NP I) (VP saw	(NP the	-5.1	1
(S (NP I) (VP saw	(NP (NP the	-6.3	4
(S (NP I) (VP saw	(S (NP the	-5.8	2
(S (NP I) (VP saw	(S (NP (NP the	-7.2	×
(S (NP I) (VP saw	(SBAR (NP the	-6.2	3
(S (NP I) (VP saw	(SBAR (NP (NP the	-7.8	×

A “word-synchronous” beam, beam size=4

Some options for neural generalization

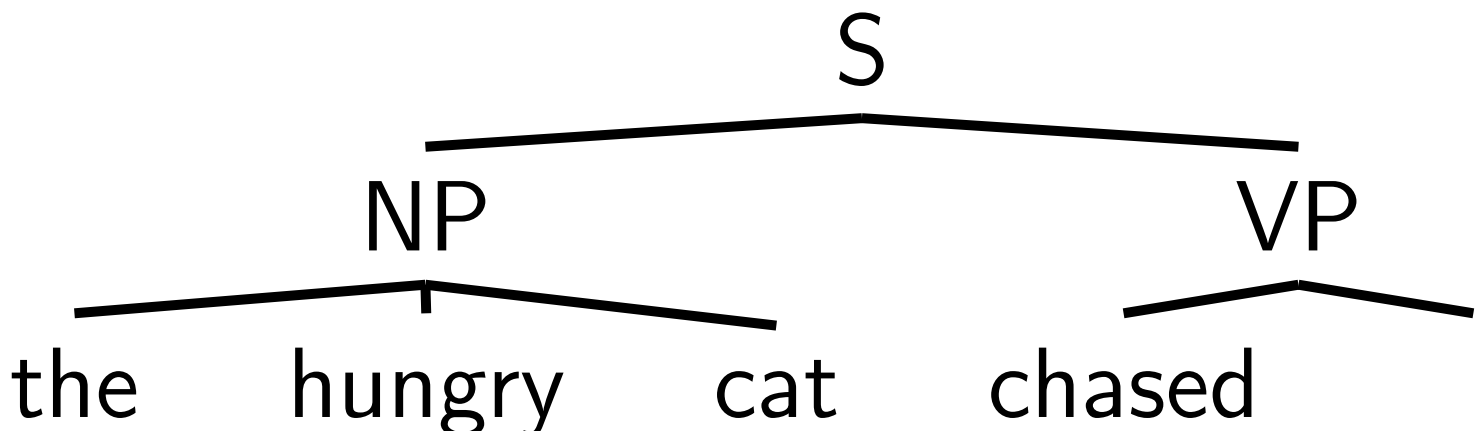
- Option 3: run RNN over *stack elements*

Action	Stack				
NT(S)	(S				
NT(NP)	(S (NP	the	hungry	cat	
Gen(the)	(S (NP the				
Gen(hungry)	(S (NP the hungry				
Gen(cat)	(S (NP the hungry cat				
REDUCE	(S (NP the hungry cat)				
NT(VP)	(S (NP the hungry cat) (VP				
Gen(chased)	(S (NP the hungry cat) (VP chased				

Generalize from the stack!

Some options for neural generalization

- Option 3: run RNN over *stack elements*

					
Action	Stack				
NT(S)	(S				
NT(NP)	(S (NP	the	hungry	cat	chased
Gen(the)	(S (NP the				
Gen(hungry)	(S (NP the hungry				
Gen(cat)	(S (NP the hungry cat				
REDUCE	(S (NP the hungry cat)				
NT(VP)	(S (NP the hungry cat) (VP				
Gen(chased)	(S (NP the hungry cat) (VP chased				

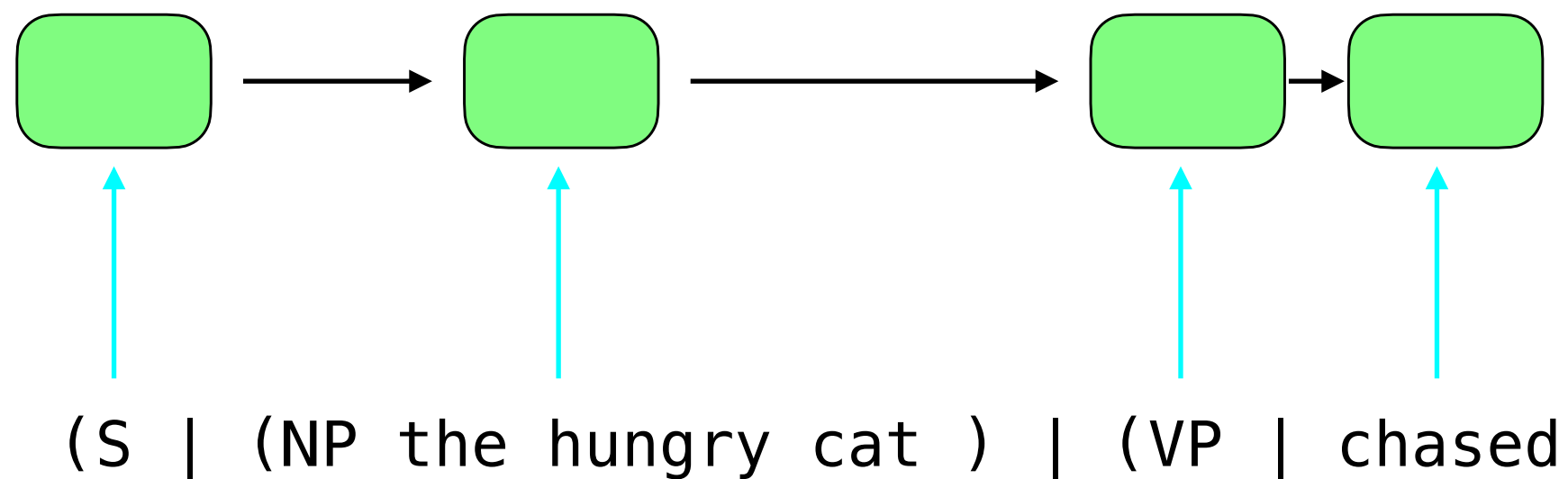
Generalize from the stack!

...but how???

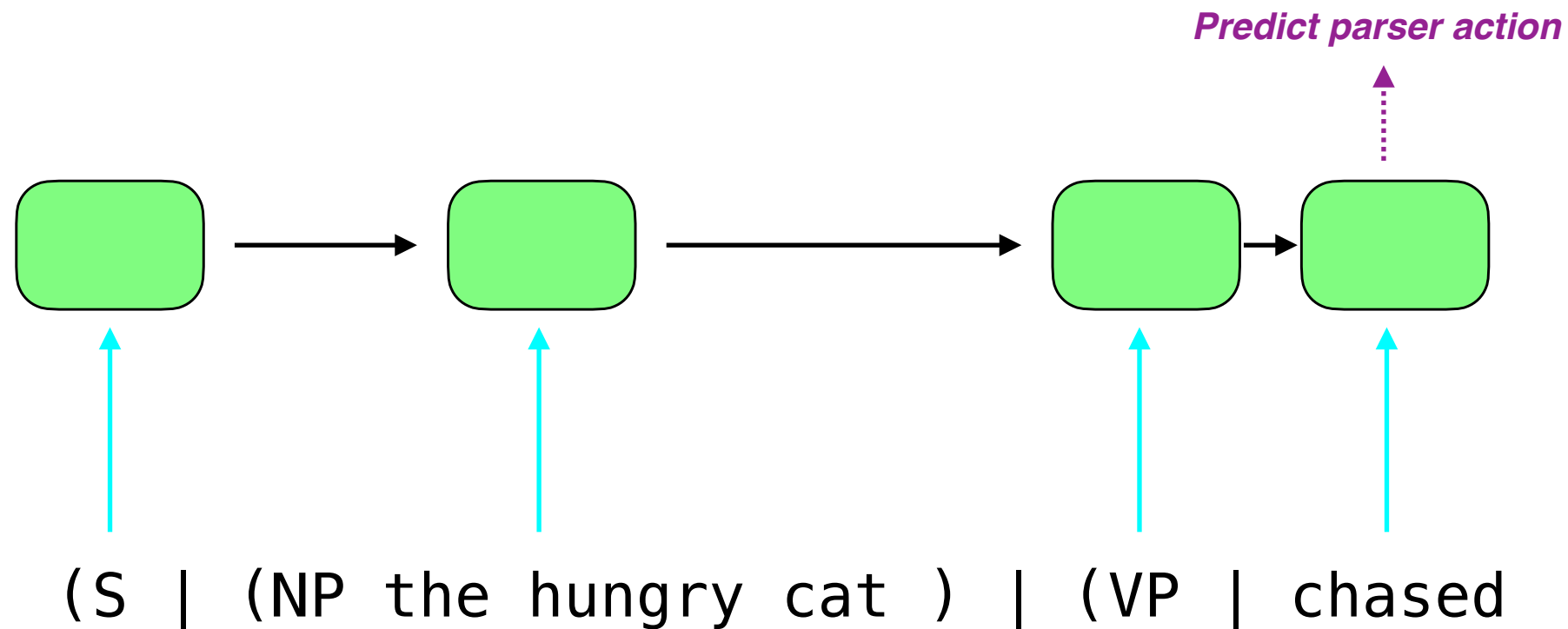
A challenge for generalization

(S | (NP the hungry cat) | (VP | chased

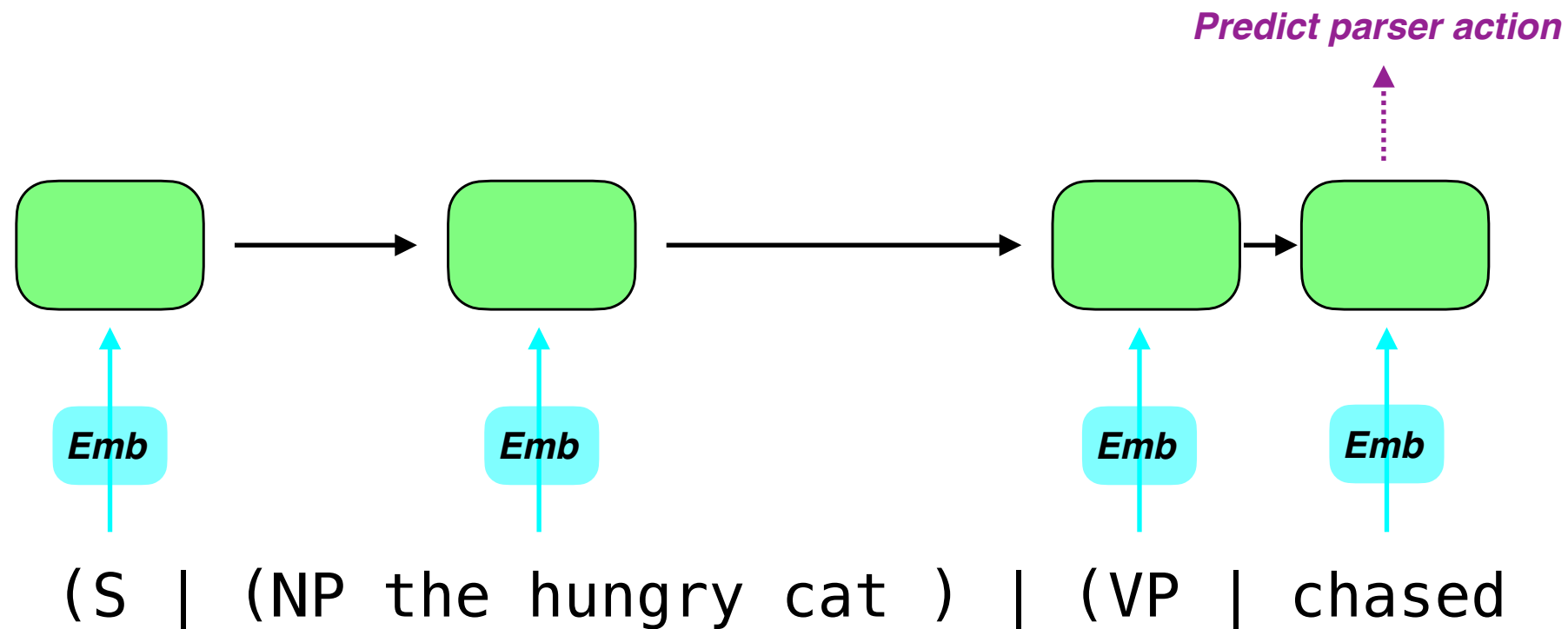
A challenge for generalization



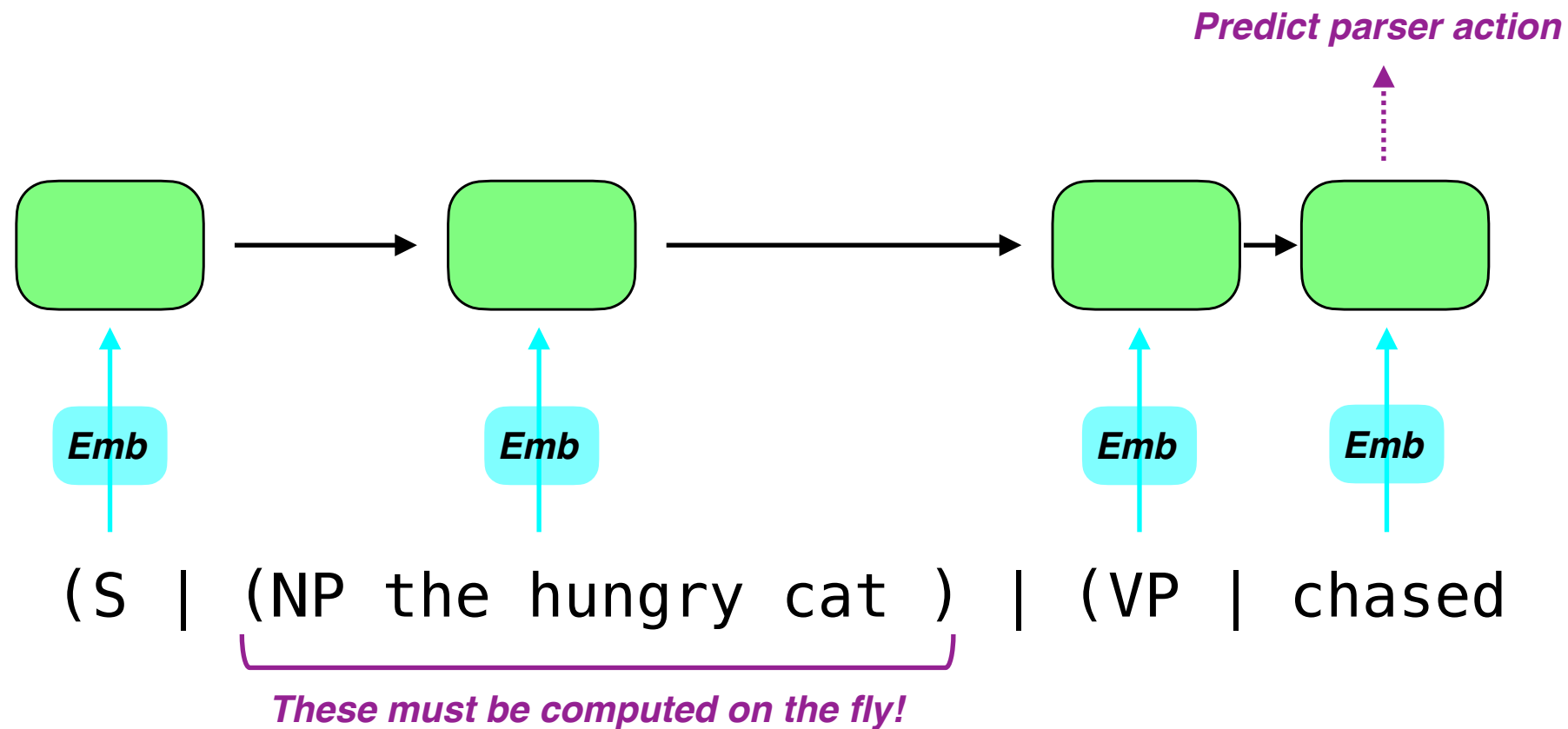
A challenge for generalization



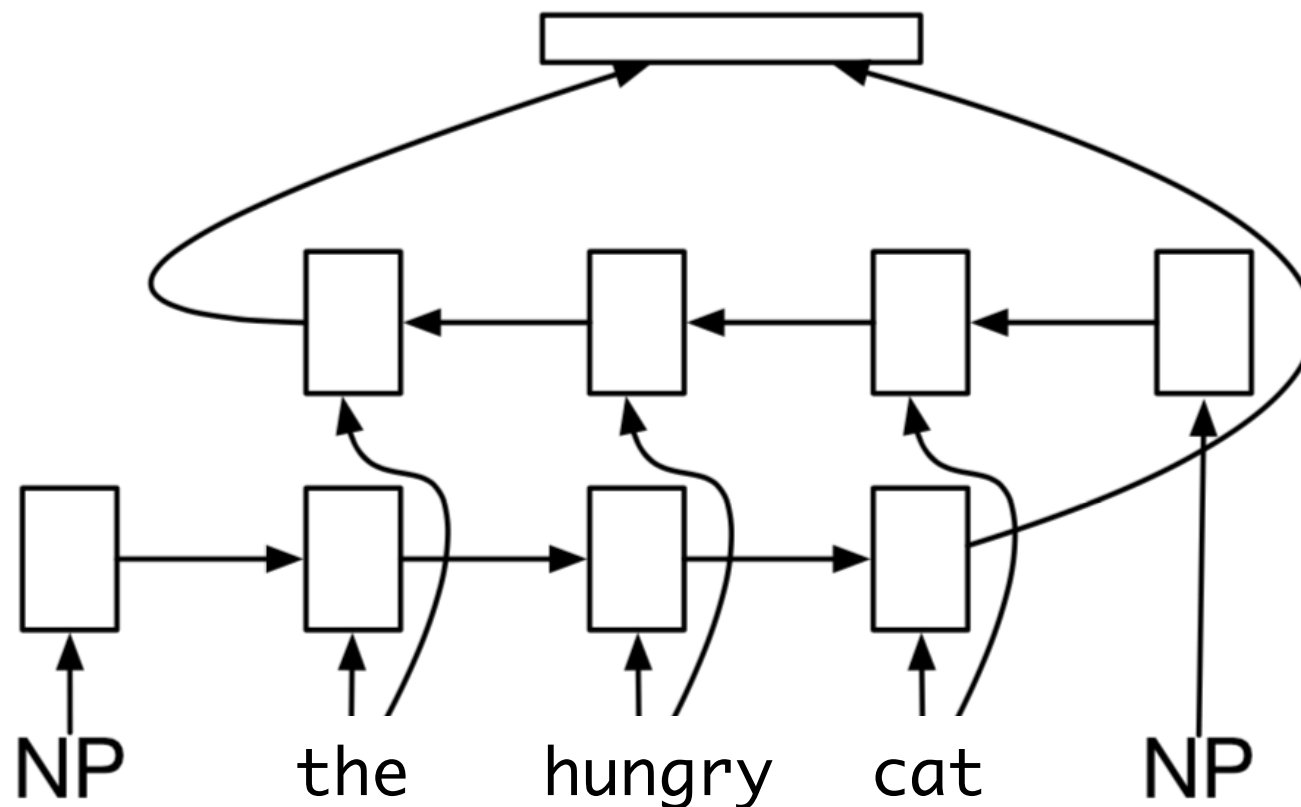
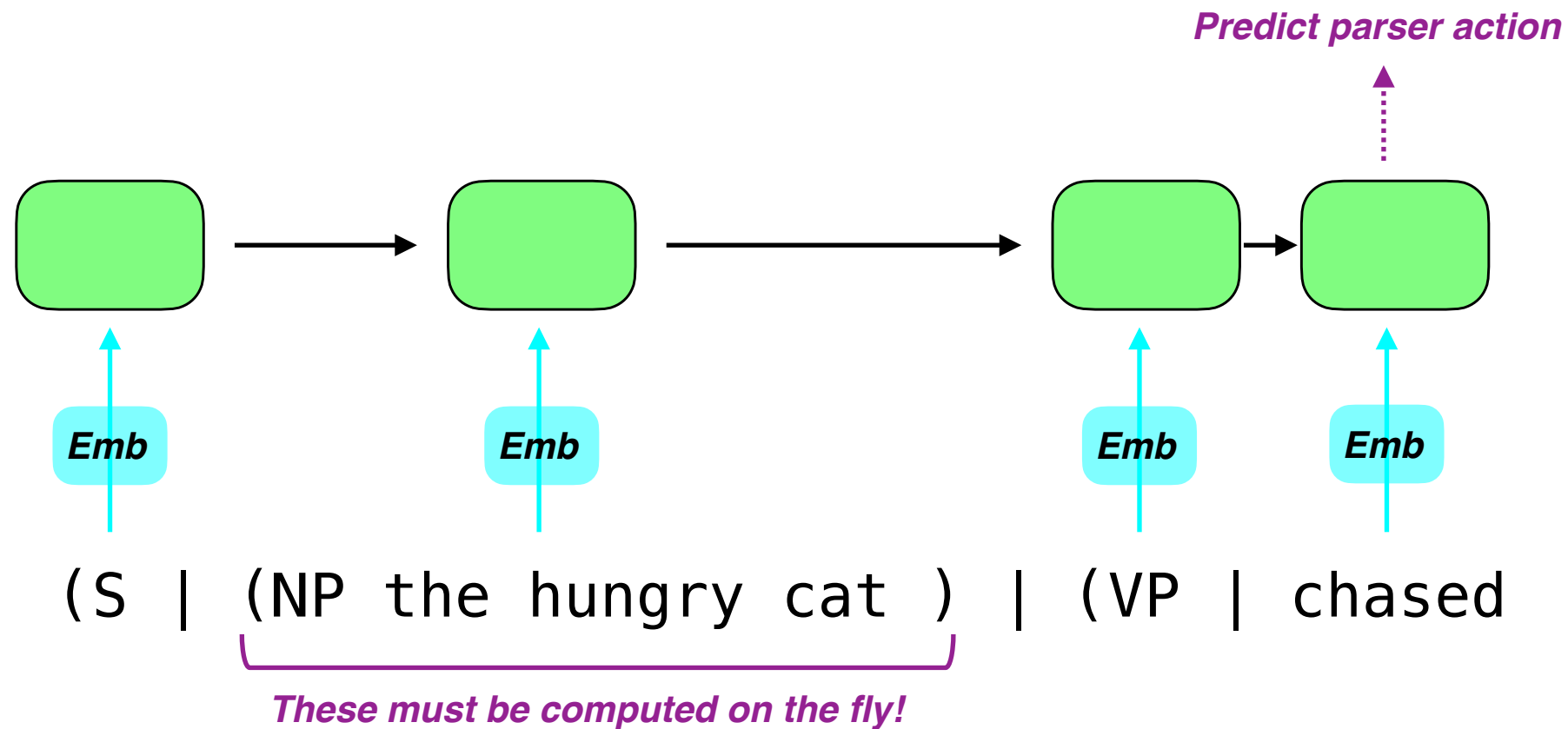
A challenge for generalization



A challenge for generalization

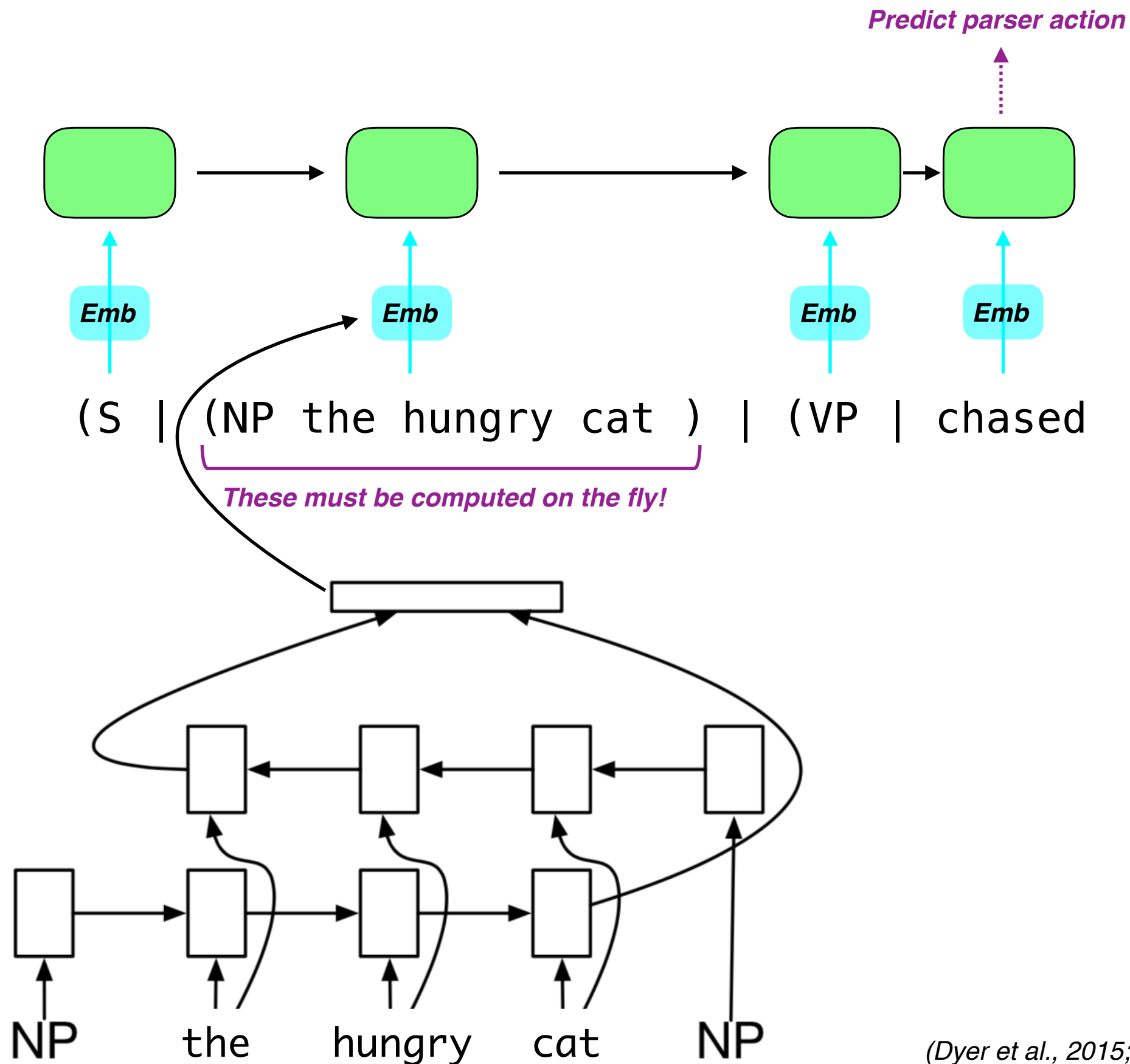


A challenge for generalization



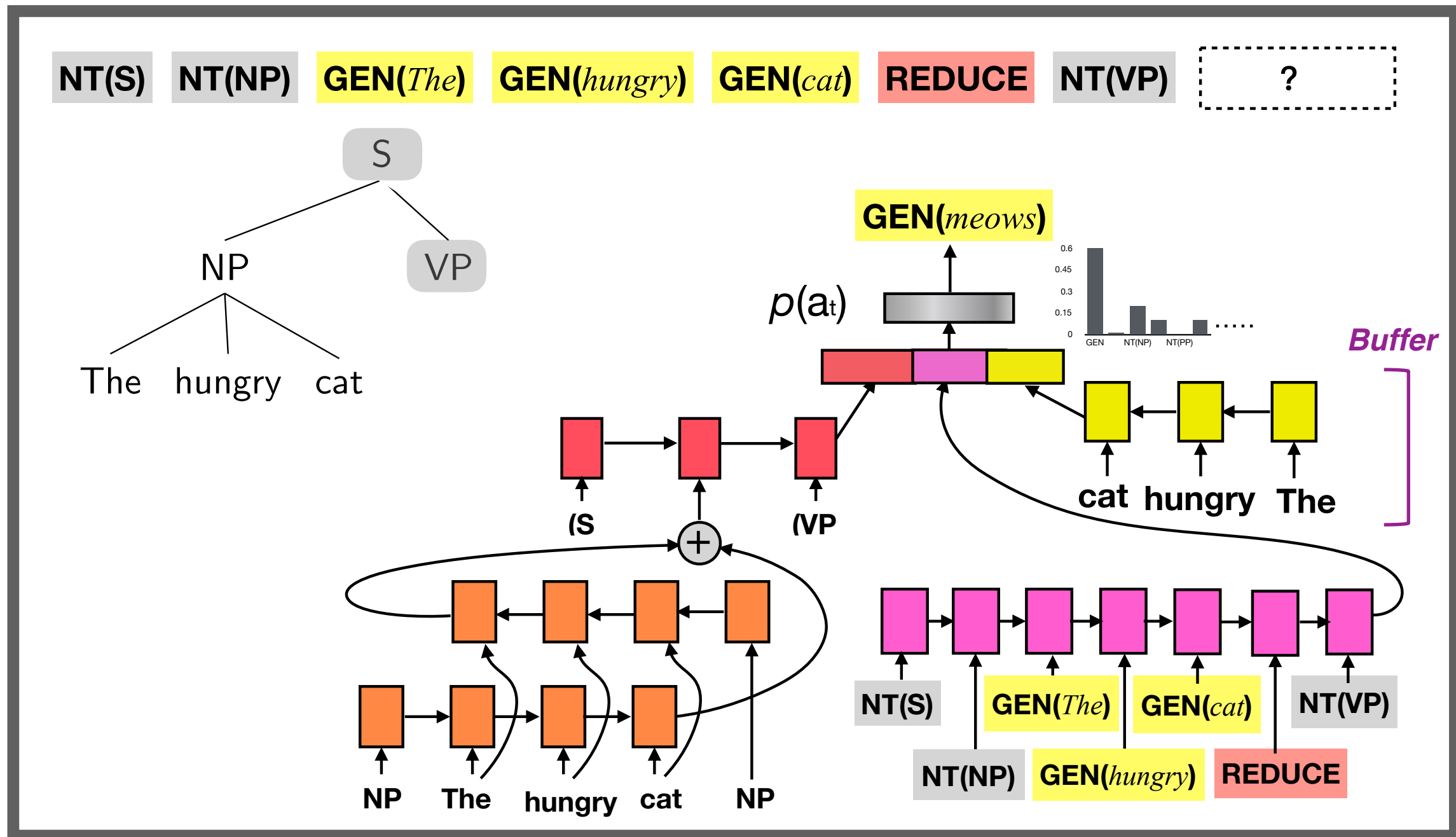
(Dyer et al., 2015; Dyer et al., 2016)

A challenge for generalization



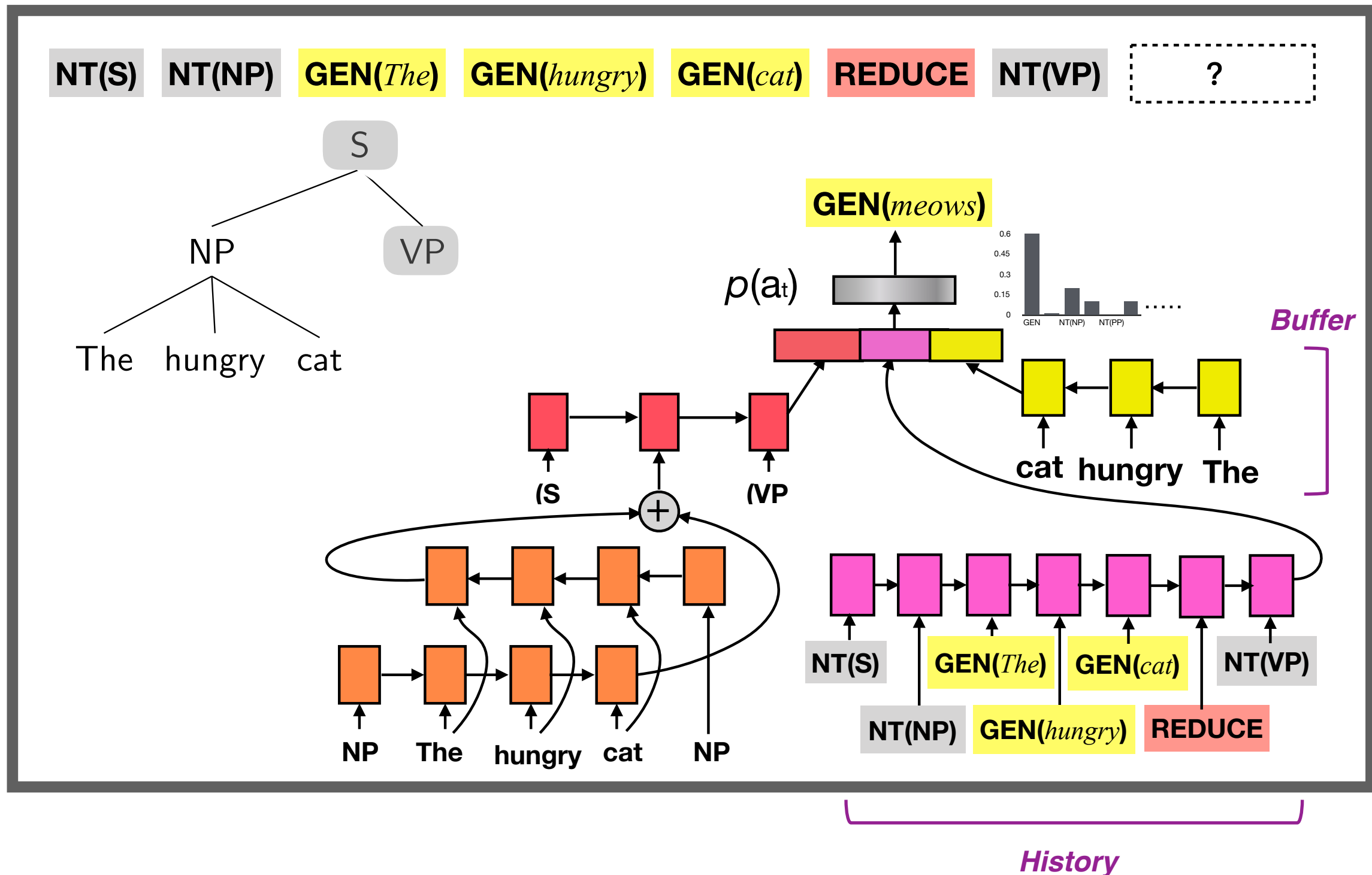
Neural generalization from *all three* sources

Recurrent Neural Network Grammars (RNNGs)



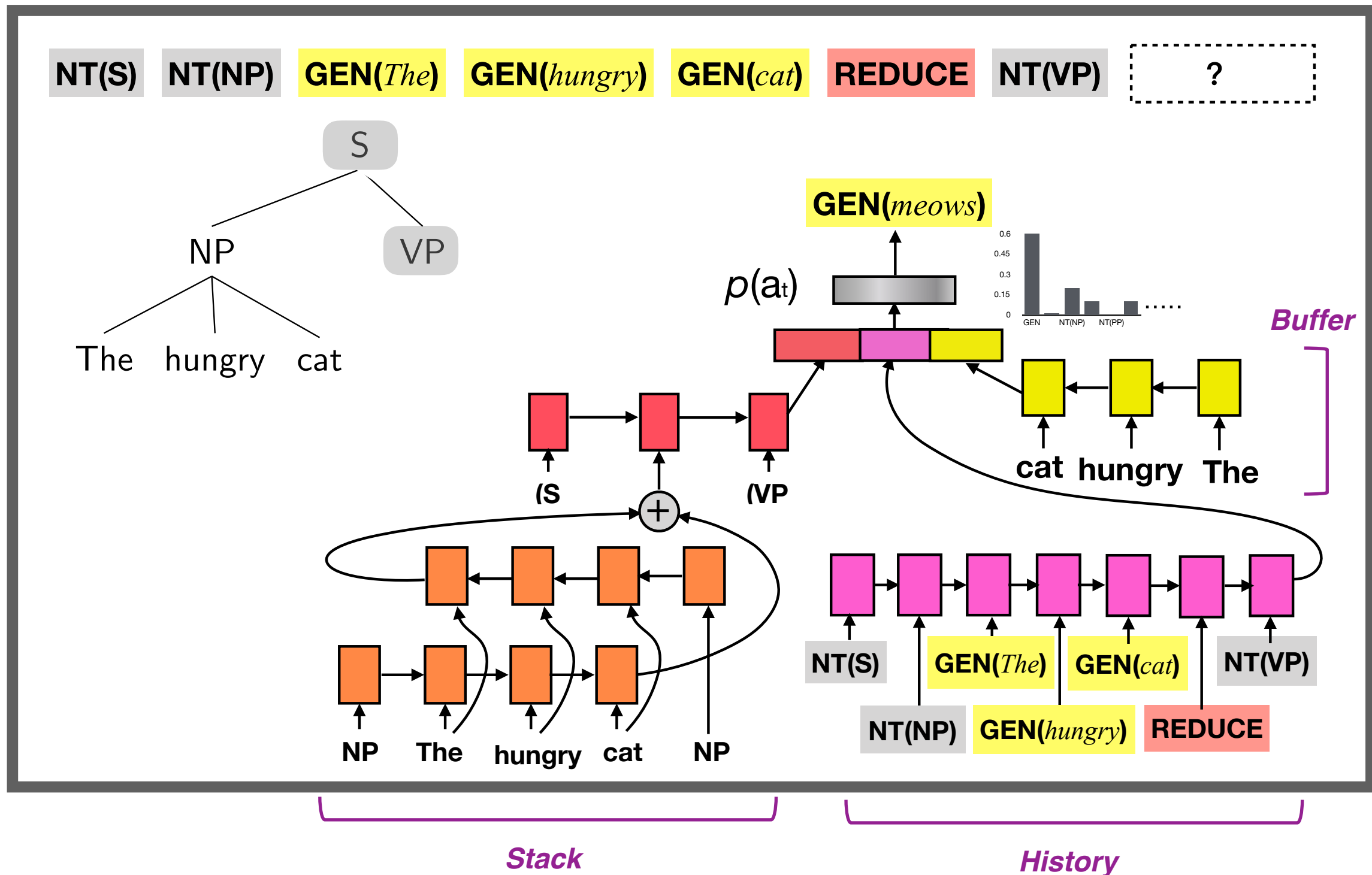
Neural generalization from *all three* sources

Recurrent Neural Network Grammars (RNNGs)



Neural generalization from *all three* sources

Recurrent Neural Network Grammars (RNNGs)



NN Language Models Tested

Model	Architecture	Training data	Data size (tokens)	Reference
JRNN	LSTM	One Billion Word	~ 800 million	Jozefowicz et al. (2016)
GRNN	LSTM	Wikipedia	~ 90 million	Gulordava et al. (2018)
RNNG	RNN Grammar	Penn Treebank	~ 1 million	Dyer et al. (2016)
TinyLSTM	LSTM	Penn Treebank	~ 1 million	—

- LSTMs have no explicit syntactic state representations.
- RNN Grammars do, but it is not always clear how they use them in making predictions.

Simplest syntactic hierarchy: subordination

Simplest syntactic hierarchy: subordination

The doctor studied the textbook .

Simplest syntactic hierarchy: subordination

✓ *The doctor studied the textbook .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
As the doctor studied the textbook .

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*

*The doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*
- As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*
- ✓ *As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook* .
- ✗ *As the doctor studied the textbook* .
- ? *The doctor studied the textbook
 , the nurse walked into the office* .
- ✓ *As the doctor studied the textbook
 , the nurse walked into the office* .

Simplest syntactic hierarchy: subordination

Context

- ✓ *The doctor studied the textbook* .
- ✗ *As the doctor studied the textbook* .
- ? *The doctor studied the textbook
, the nurse walked into the office* .
- ✓ *As the doctor studied the textbook
, the nurse walked into the office* .

Simplest syntactic hierarchy: subordination

- ✓ *The doctor studied the textbook* .
- ✗ *As the doctor studied the textbook* .
- ? *The doctor studied the textbook
, the nurse walked into the office* .
- ✓ *As the doctor studied the textbook
, the nurse walked into the office* .

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

✓ *The doctor studied the textbook* | .

Context | Completion

✗ *As the doctor studied the textbook* .

? *The doctor studied the textbook
, the nurse walked into the office* .

✓ *As the doctor studied the textbook
, the nurse walked into the office* .

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*
- ✓ *As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*
- ✓ *As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

“No-matrix” variants
(No subsequent matrix clause)

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- ? *The doctor studied the textbook
, the nurse walked into the office .*
- ✓ *As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

“No-matrix” variants

(No subsequent matrix clause)

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*

“Matrix” variants

(There is a subsequent matrix clause)

- ? *The doctor studied the textbook
, the nurse walked into the office .*
- ✓ *As the doctor studied the textbook
, the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

“No-matrix” variants

(No subsequent matrix clause)

- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- Surprisal difference (should be positive)

“Matrix” variants

(There is a subsequent matrix clause)

- ? *The doctor studied the textbook , the nurse walked into the office .*
- ✓ *As the doctor studied the textbook , the nurse walked into the office .*

Simplest syntactic hierarchy: subordination

$$-\log P(\text{Completion}|\text{Context})$$

“No-matrix” variants

(No subsequent matrix clause)

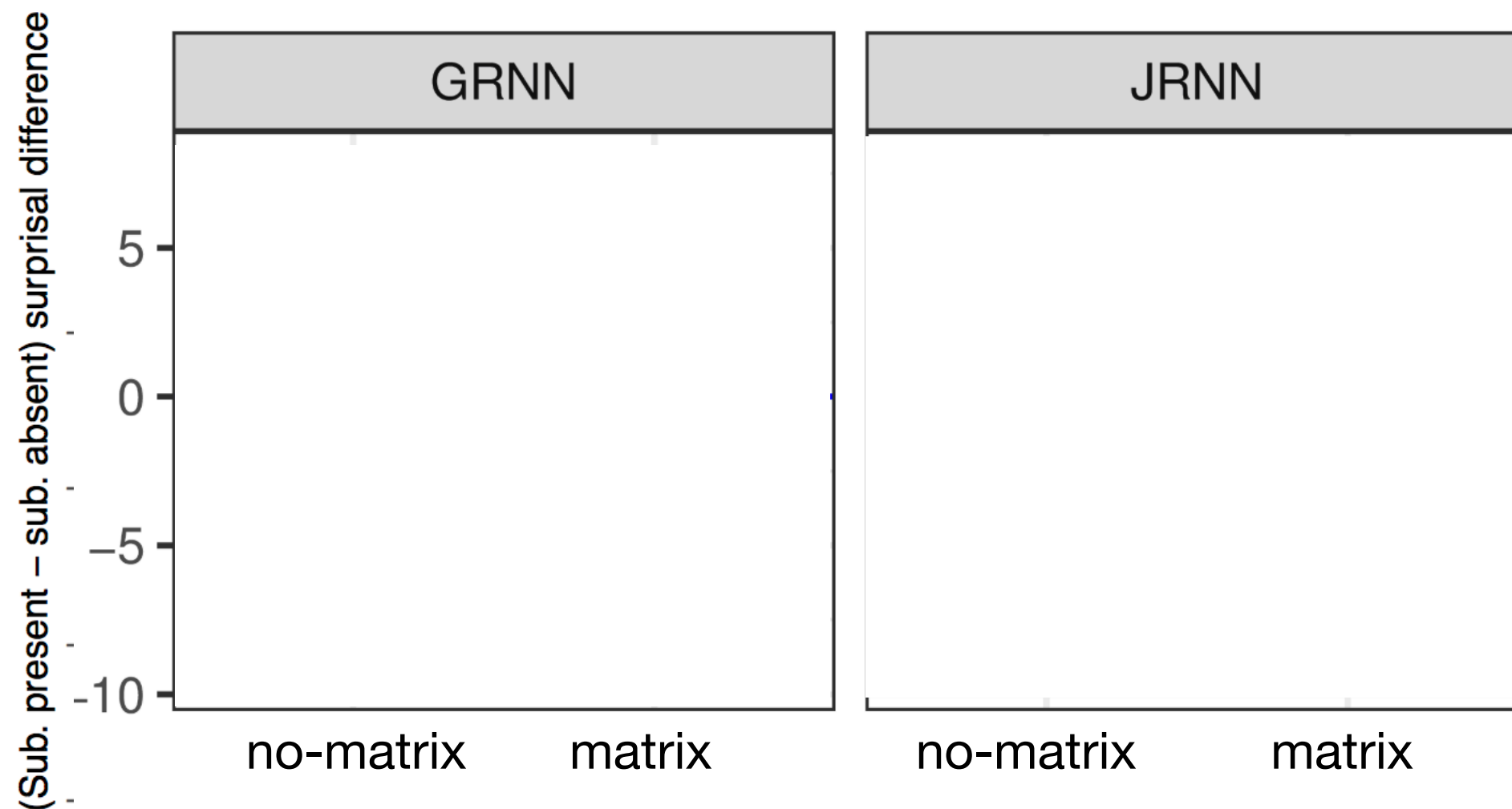
- ✓ *The doctor studied the textbook .*
- ✗ *As the doctor studied the textbook .*
- Surprisal difference (should be positive)

“Matrix” variants

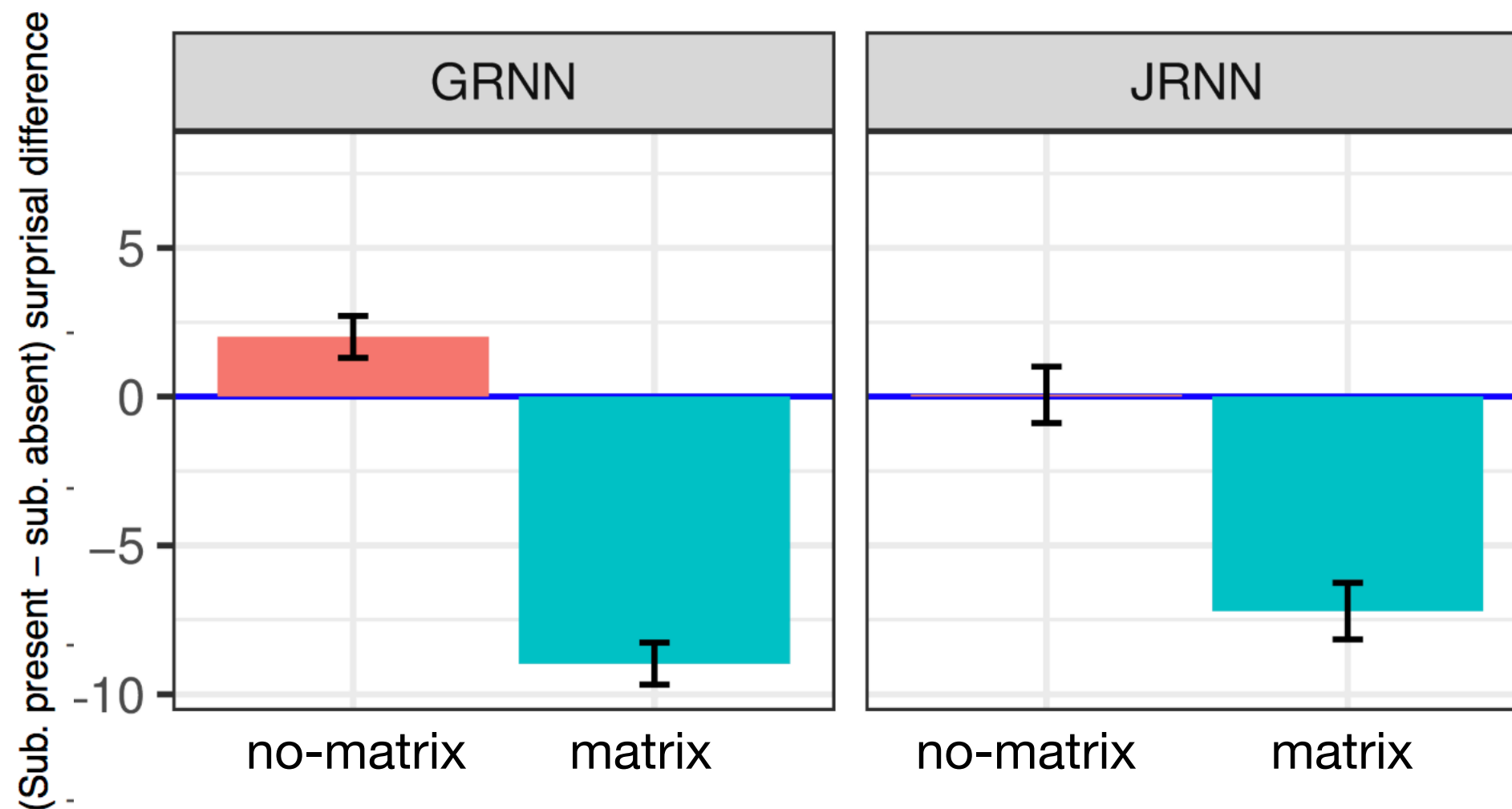
(There is a subsequent matrix clause)

- ? *The doctor studied the textbook , the nurse walked into the office .*
- ✓ *As the doctor studied the textbook , the nurse walked into the office .*
- Surprisal difference (should be negative)

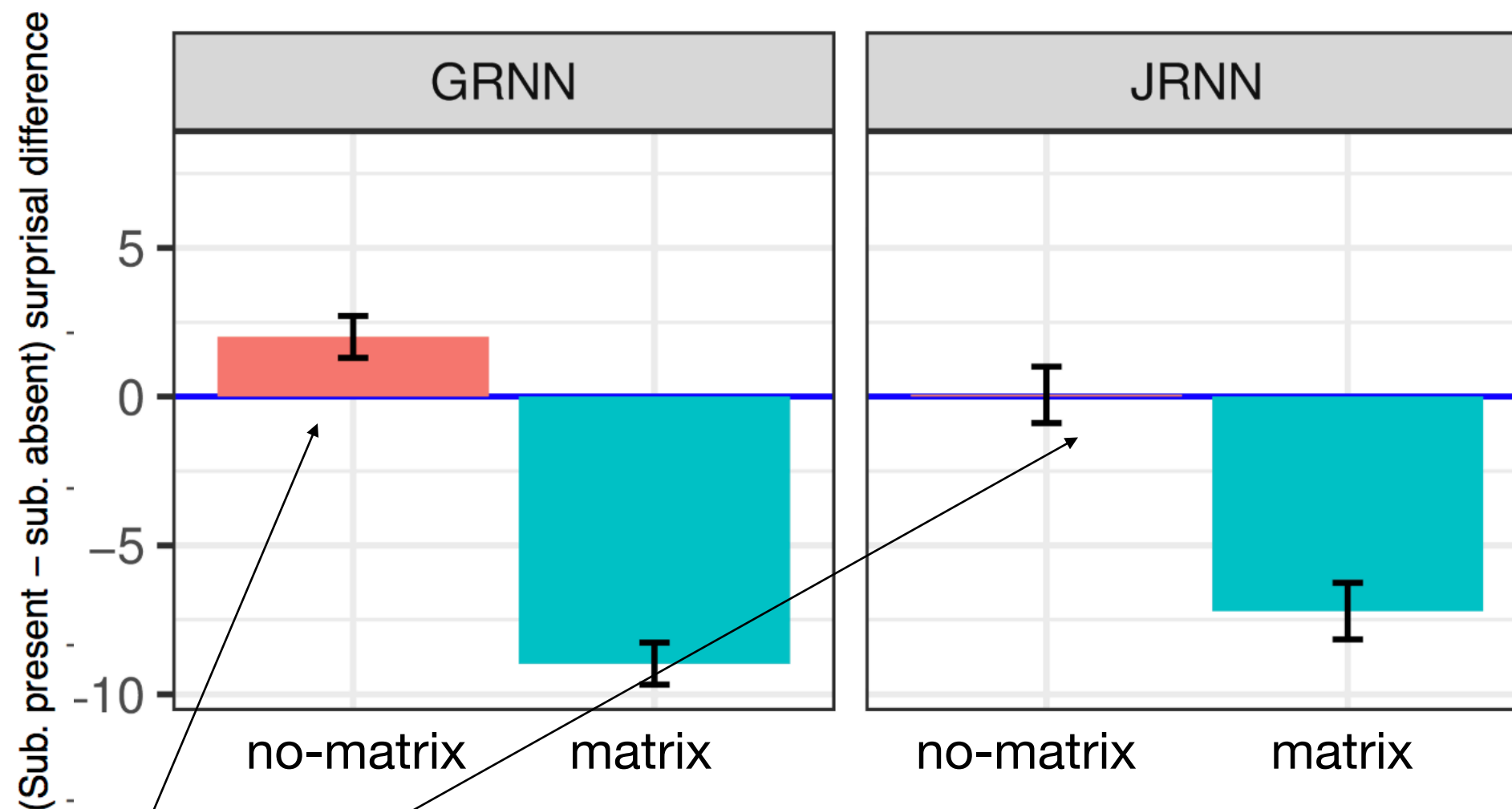
Effects of Subordinate Clauses



Effects of Subordinate Clauses

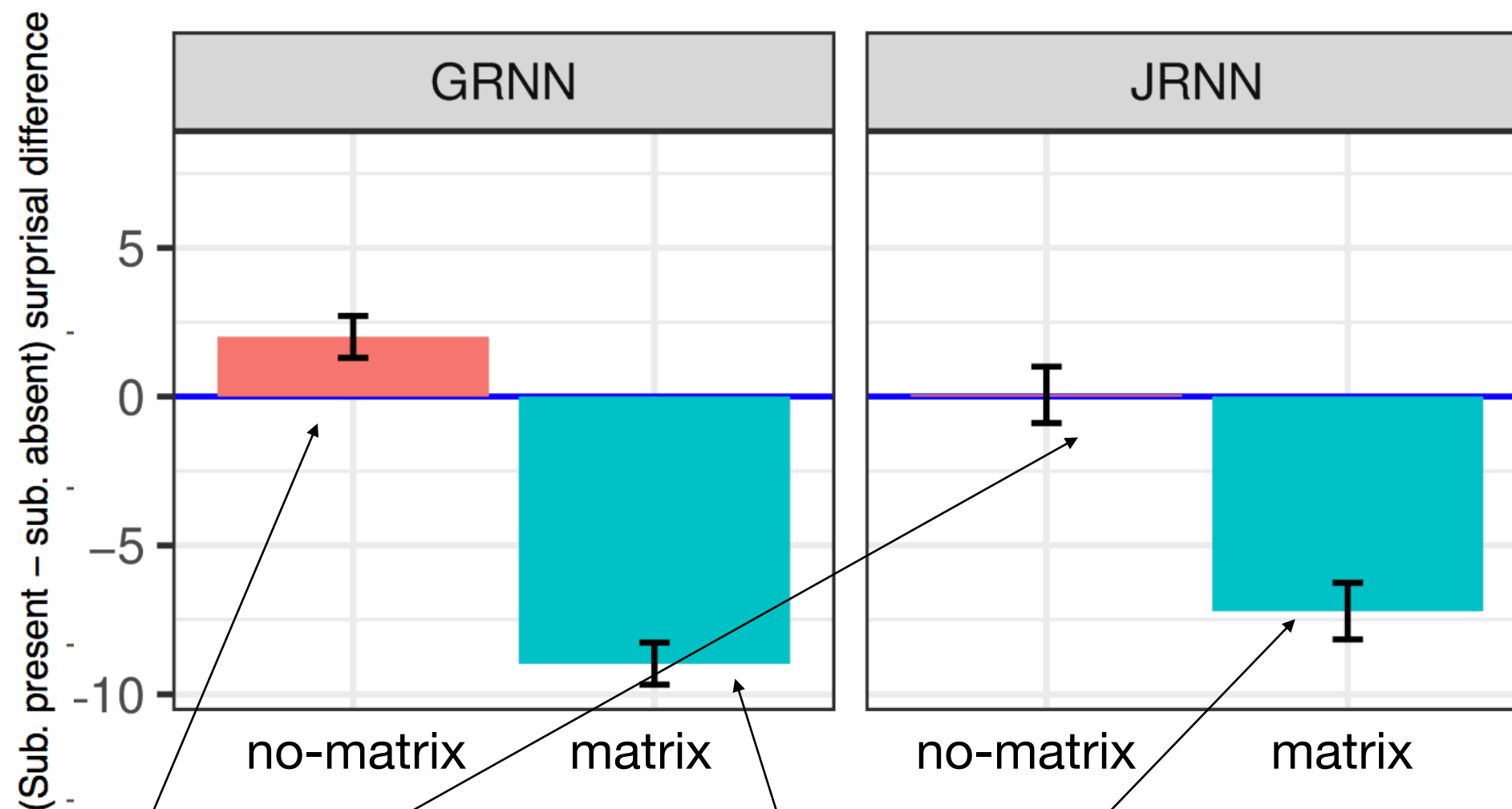


Effects of Subordinate Clauses



No-matrix penalty

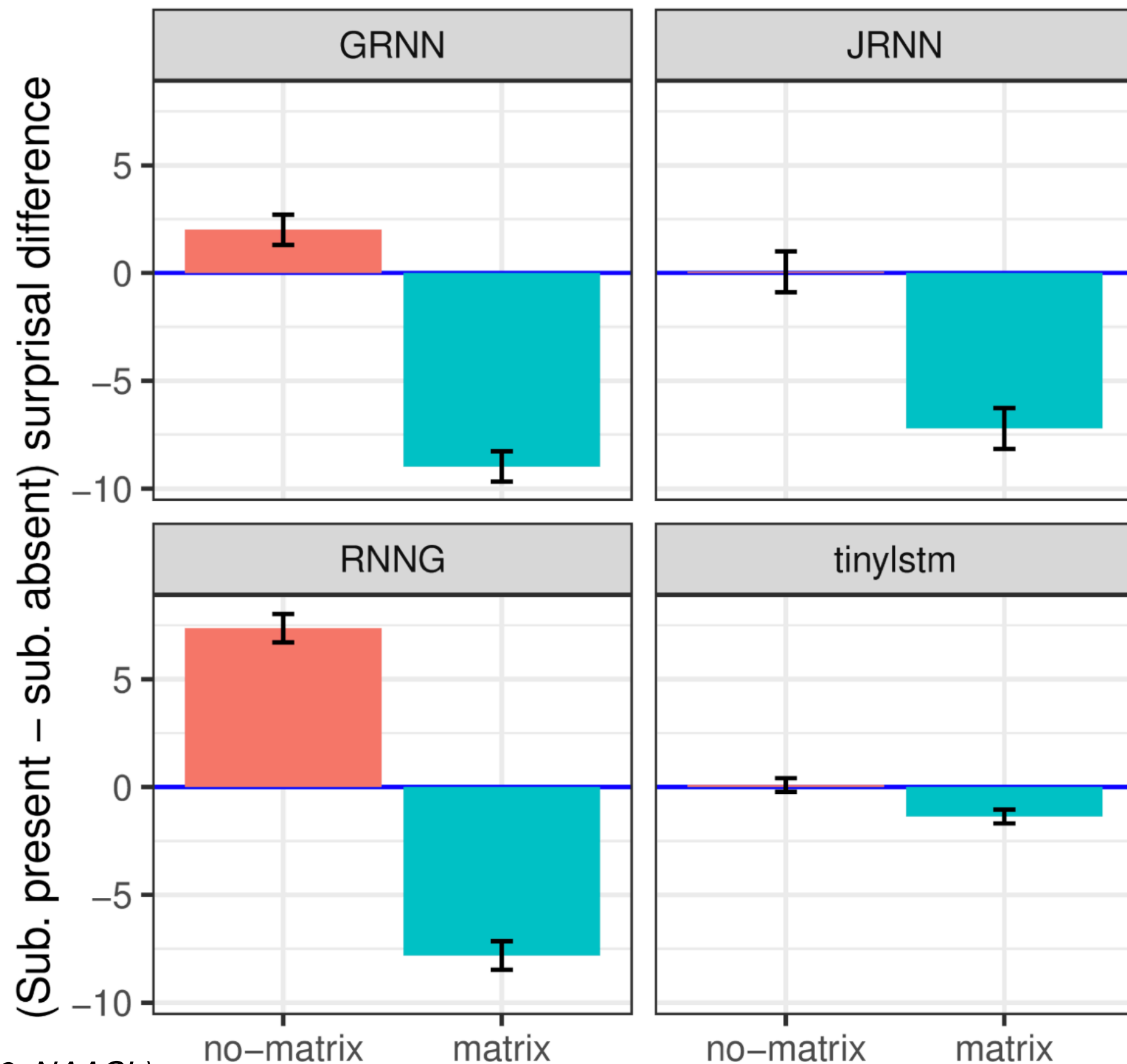
Effects of Subordinate Clauses



No-matrix penalty

Matrix licensing

Subordination: results



Subordination: summary

- All models learned *something* about the contingency between initial subordinator & need for a second clause
- Explicit representation of grammatical structure substantially sharpened that contingency

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

When the dog scratched, the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

When the dog scratched, the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

When the dog scratched, the vet ~~with~~ his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

When the dog scratched, the vet with his new assistant removed the muzzle.



Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.


difficulty here

When the dog scratched, the vet ~~with~~ his new assistant removed the muzzle.


easier

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

↑
difficulty here

When the dog scratched, the vet with his new assistant removed the muzzle.

↑
easier

When the dog arrived the vet with his new assistant removed the muzzle.

When the dog arrived, the vet with his new assistant removed the muzzle.

Garden-pathing

When the dog scratched the vet with his new assistant removed the muzzle.

When the dog scratched, the vet with his new assistant removed the muzzle.

When the dog arrived the vet with his new assistant removed the muzzle.

When the dog arrived, the vet with his new assistant removed the muzzle.

Garden-pathing

(a) [transitive, -comma]

When the dog scratched the vet with his new assistant removed the muzzle.

(b) [transitive, +comma]

When the dog scratched, the vet with his new assistant removed the muzzle.

(c) [intransitive, -comma]

When the dog arrived the vet with his new assistant removed the muzzle.

(d) [intransitive, +comma]

When the dog arrived, the vet with his new assistant removed the muzzle.

Garden-pathing

(a) [transitive, -comma]

When the dog scratched the vet with his new assistant removed the muzzle.

(b) [transitive, +comma]

When the dog scratched, the vet with his new assistant removed the muzzle.

(c) [intransitive, -comma]

When the dog arrived the vet with his new assistant removed the muzzle.

(d) [intransitive, +comma]

When the dog arrived, the vet with his new assistant removed the muzzle.

Garden-pathing

(a) [transitive, -comma]

When the dog scratched the vet with his new assistant removed the muzzle.

(b) [transitive, +comma]

When the dog scratched, the vet with his new assistant removed the muzzle.

(c) [intransitive, -comma]

When the dog arrived the vet with his new assistant removed the muzzle.

(d) [intransitive, +comma]

When the dog arrived, the vet with his new assistant removed the muzzle.

$$S(x) = -\log P(\text{removed} | \text{Context of version } x)$$

Garden-pathing

(a) [transitive, -comma]

When the dog scratched the vet with his new assistant removed the muzzle.

(b) [transitive, +comma]

When the dog scratched, the vet with his new assistant removed the muzzle.

(c) [intransitive, -comma]

When the dog arrived the vet with his new assistant removed the muzzle.

(d) [intransitive, +comma]

When the dog arrived, the vet with his new assistant removed the muzzle.

$$S(x) = -\log P(\text{removed} | \text{Context of version } x)$$

$$(i) \ S(a) > S(b)$$

Garden-pathing

(a) [transitive, -comma]

When the dog scratched the vet with his new assistant removed the muzzle.

(b) [transitive, +comma]

When the dog scratched, the vet with his new assistant removed the muzzle.

(c) [intransitive, -comma]

When the dog arrived the vet with his new assistant removed the muzzle.

(d) [intransitive, +comma]

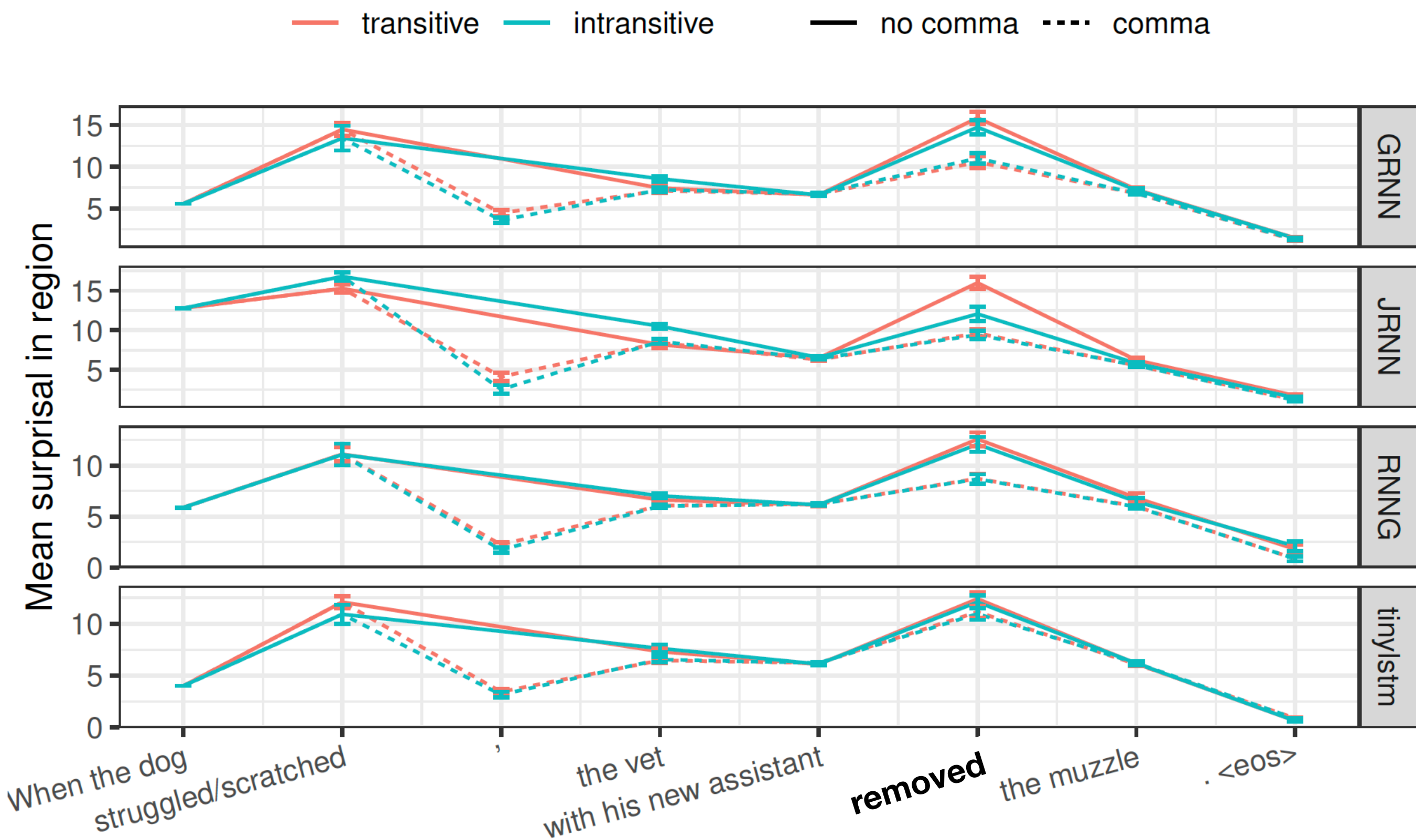
When the dog arrived, the vet with his new assistant removed the muzzle.

$$S(x) = -\log P(\text{removed} | \text{Context of version } x)$$

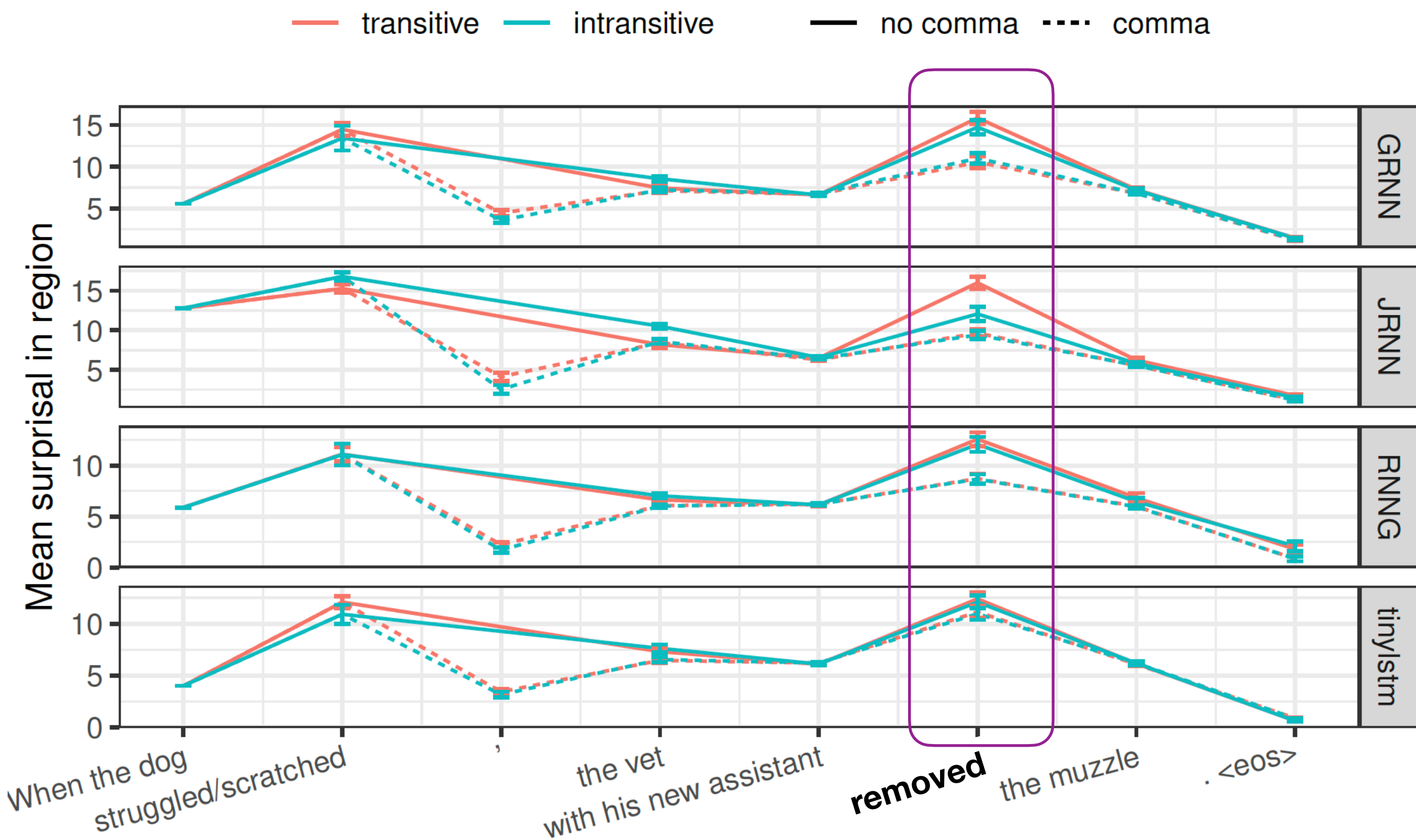
$$(i) \ S(a) > S(b)$$

$$(ii) \ S(a) - S(b) > S(c) - S(d)$$

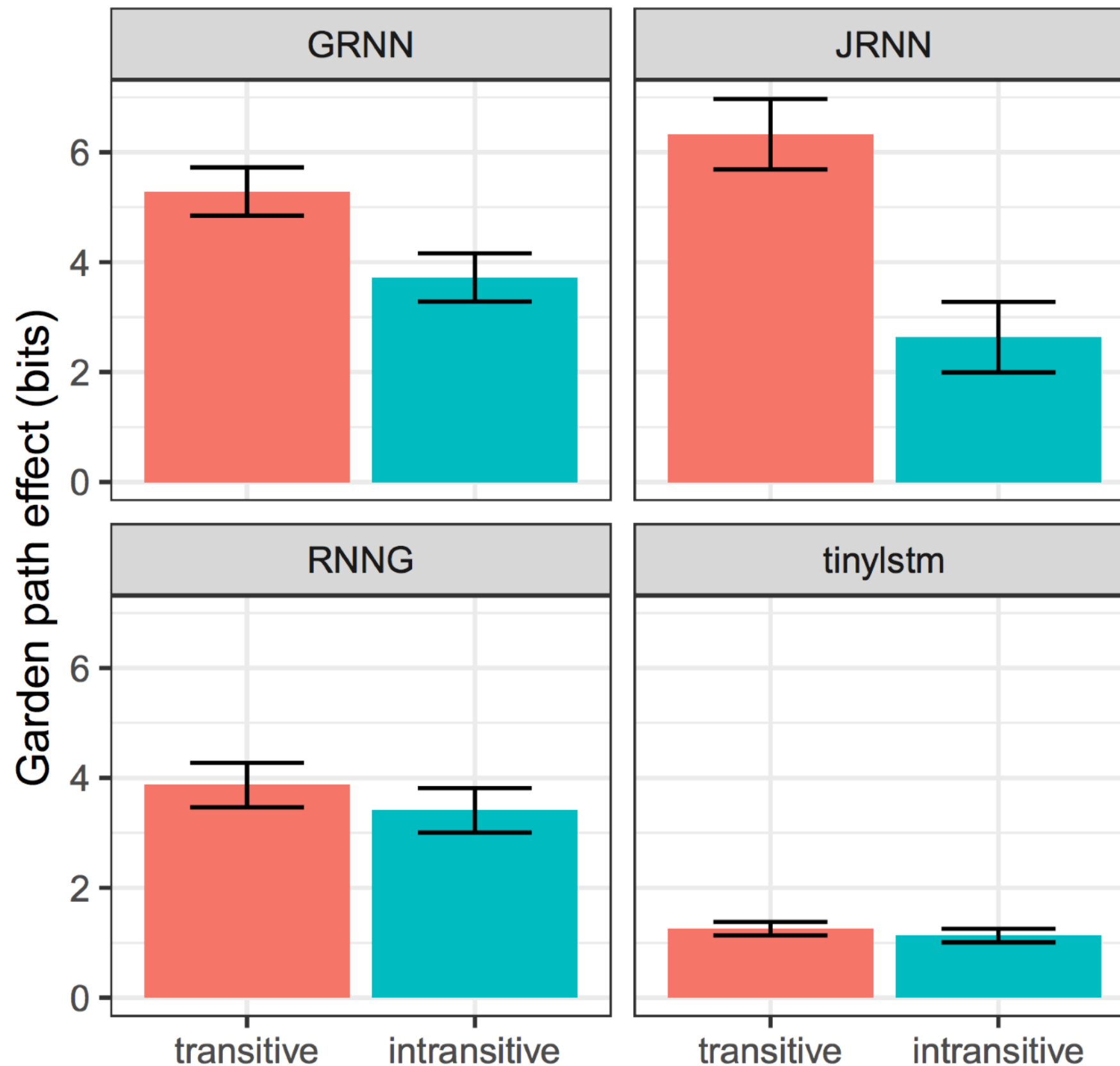
Region-by-region surprisal profiles



Region-by-region surprisal profiles



NP/Z Garden Path Results



NP/Z Garden Paths: Degradation Over Time

- (a) [short, -comma] As the author studying Babylon in ancient times wrote the book **grew**.
- (b) [short, +comma] As the author studying Babylon in ancient times wrote, the book **grew**.
- (c) [long, -comma] As the author wrote the book studying Babylon in ancient times **grew**.
- (d) [long, +comma] As the author wrote, the book studying Babylon in ancient times **grew**.

*(Warner & Glass, 1987; Ferreira & Henderson, 1991;
Tabor & Hutchins, 2004; Levy et al., 2009)*

NP/Z Garden Paths: Degradation Over Time

- (a) [short, -comma] As the author studying Babylon in ancient times wrote the book **grew**.
- (b) [short, +comma] As the author studying Babylon in ancient times wrote, the book **grew**.
- (c) [long, -comma] As the author wrote the book studying Babylon in ancient times **grew**.
- (d) [long, +comma] As the author wrote, the book studying Babylon in ancient times **grew**.

*(Warner & Glass, 1987; Ferreira & Henderson, 1991;
Tabor & Hutchins, 2004; Levy et al., 2009)*

Prediction:

$$S(a) - S(b) \approx S(c) - S(d) > 0$$

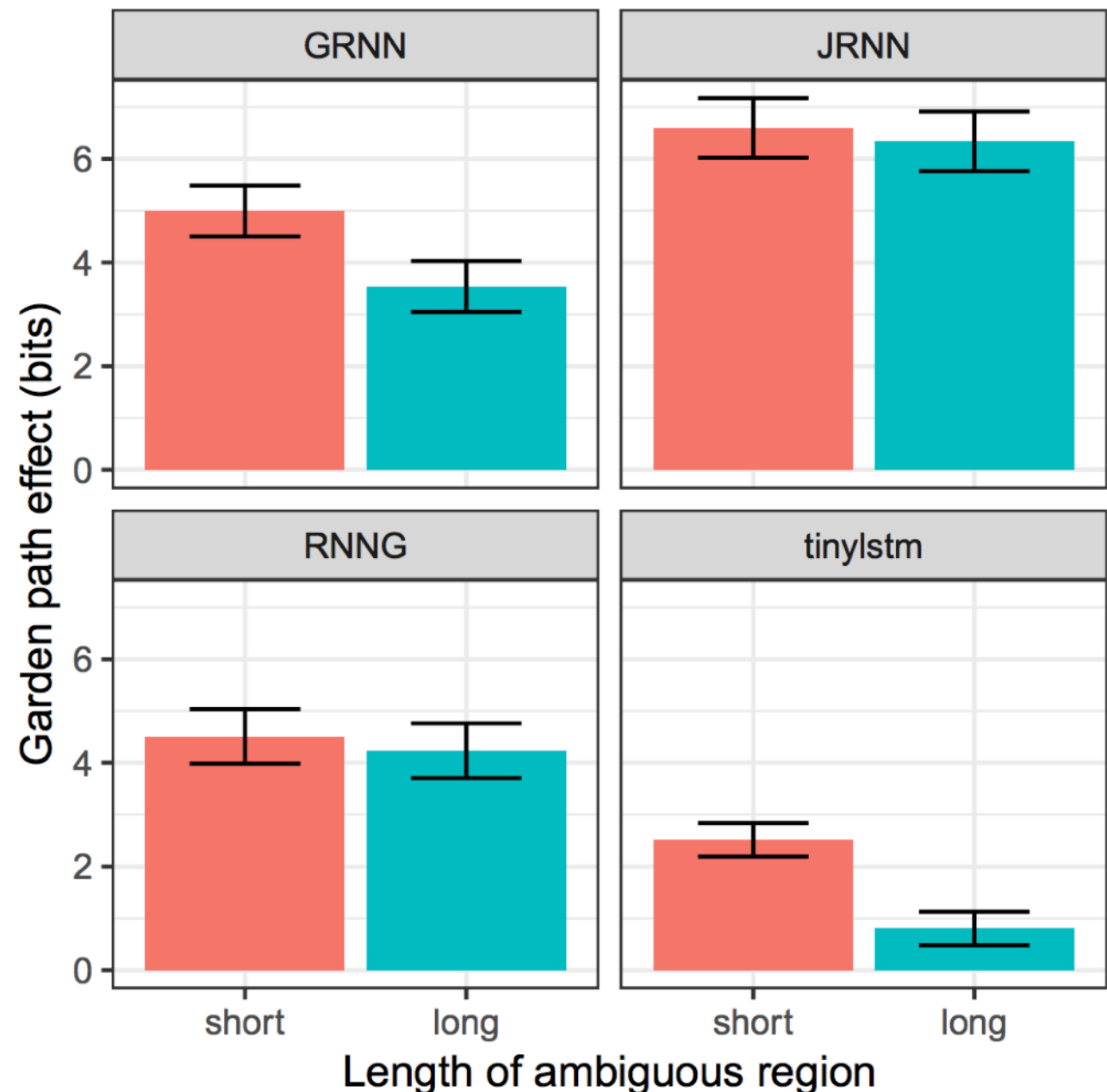
NP/Z Garden Paths: Degradation Over Time

- (a) [short, -comma] As the author studying Babylon in ancient times wrote the book **grew**.
- (b) [short, +comma] As the author studying Babylon in ancient times wrote, the book **grew**.
- (c) [long, -comma] As the author wrote the book studying Babylon in ancient times **grew**.
- (d) [long, +comma] As the author wrote, the book studying Babylon in ancient times **grew**.

(Warner & Glass, 1987; Ferreira & Henderson, 1991;
Tabor & Hutchins, 2004; Levy et al., 2009)

Prediction:

$$S(a) - S(b) \approx S(c) - S(d) > 0$$



(Futrell et al. 2019, NAACL)

"Digging in" in human NP/Z garden-pathing

△(a) [short, -object]

As the author wrote the book **grew**.

□(b) [short, +object]

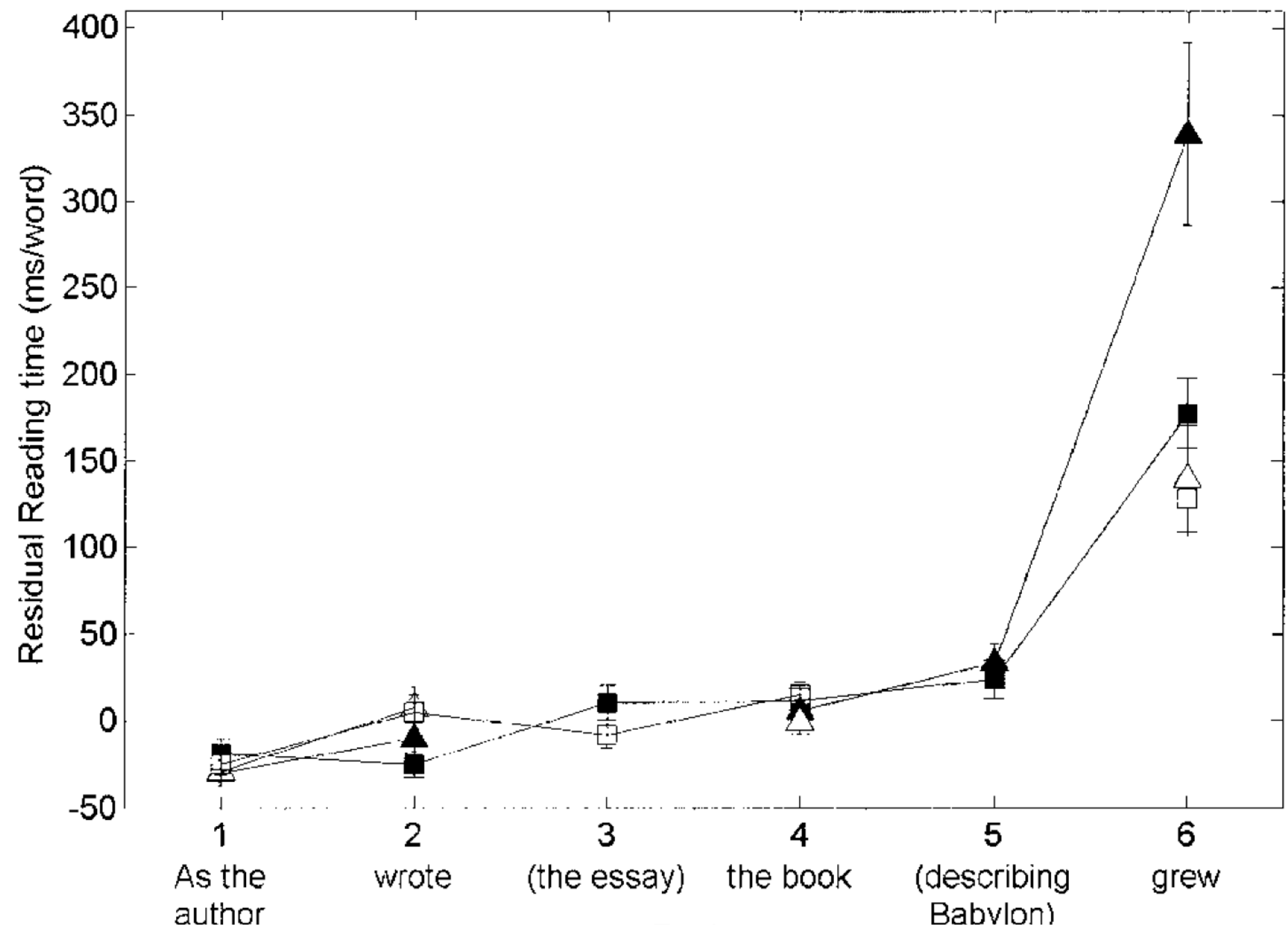
As the author wrote the essay the book **grew**.

▲(c) [long, -object]

As the author wrote the book describing Babylon **grew**.

■(d) [long, +object]

As the author wrote the essay the book describing Babylon **grew**.



"Digging in" in human NP/Z garden-pathing

△(a) [short, -object]

As the author wrote the book **grew**.

□(b) [short, +object]

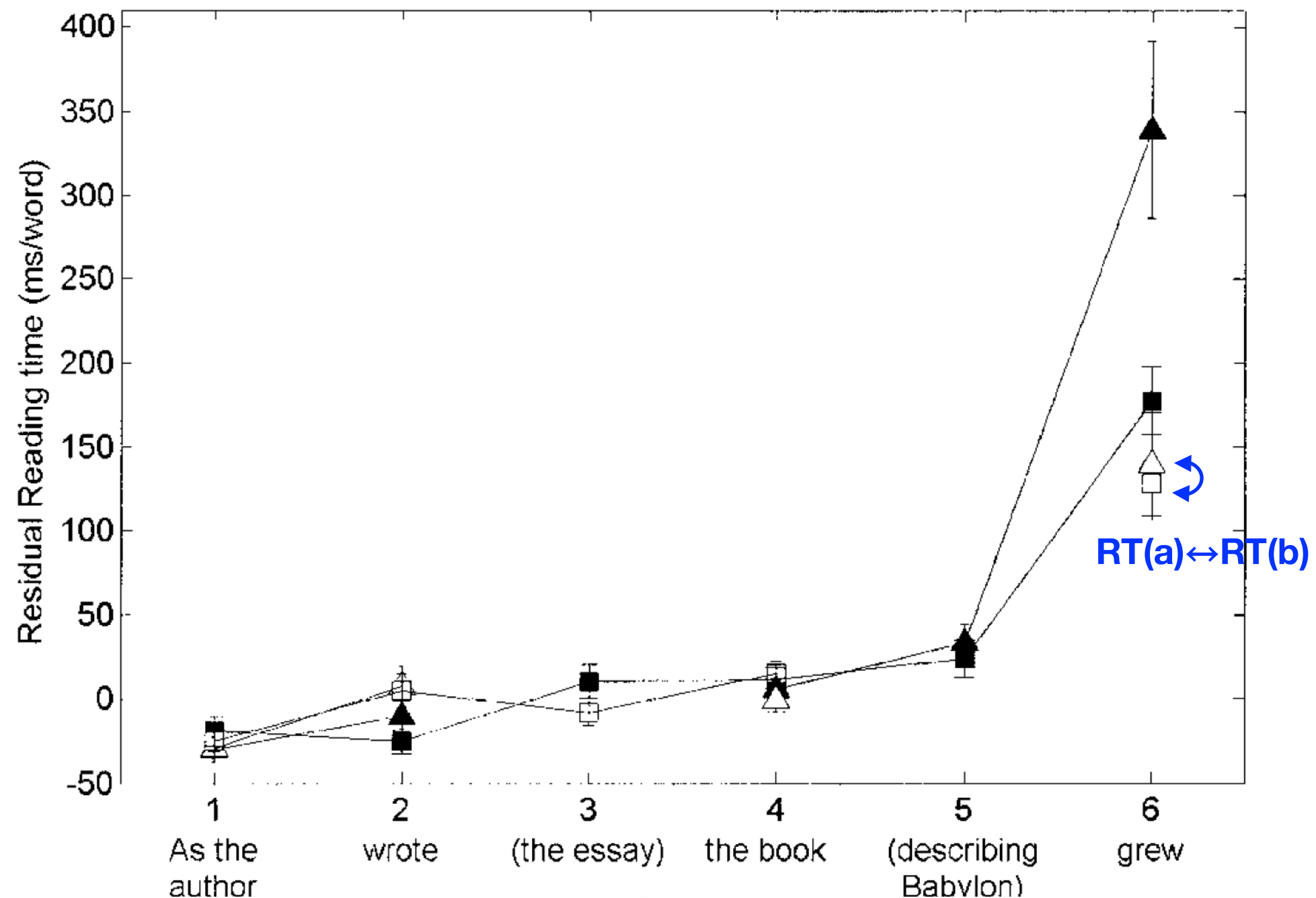
As the author wrote the essay the book **grew**.

▲(c) [long, -object]

As the author wrote the book describing Babylon **grew**.

■(d) [long, +object]

As the author wrote the essay the book describing Babylon **grew**.



"Digging in" in human NP/Z garden-pathing

△(a) [short, -object]

As the author wrote the book **grew**.

□(b) [short, +object]

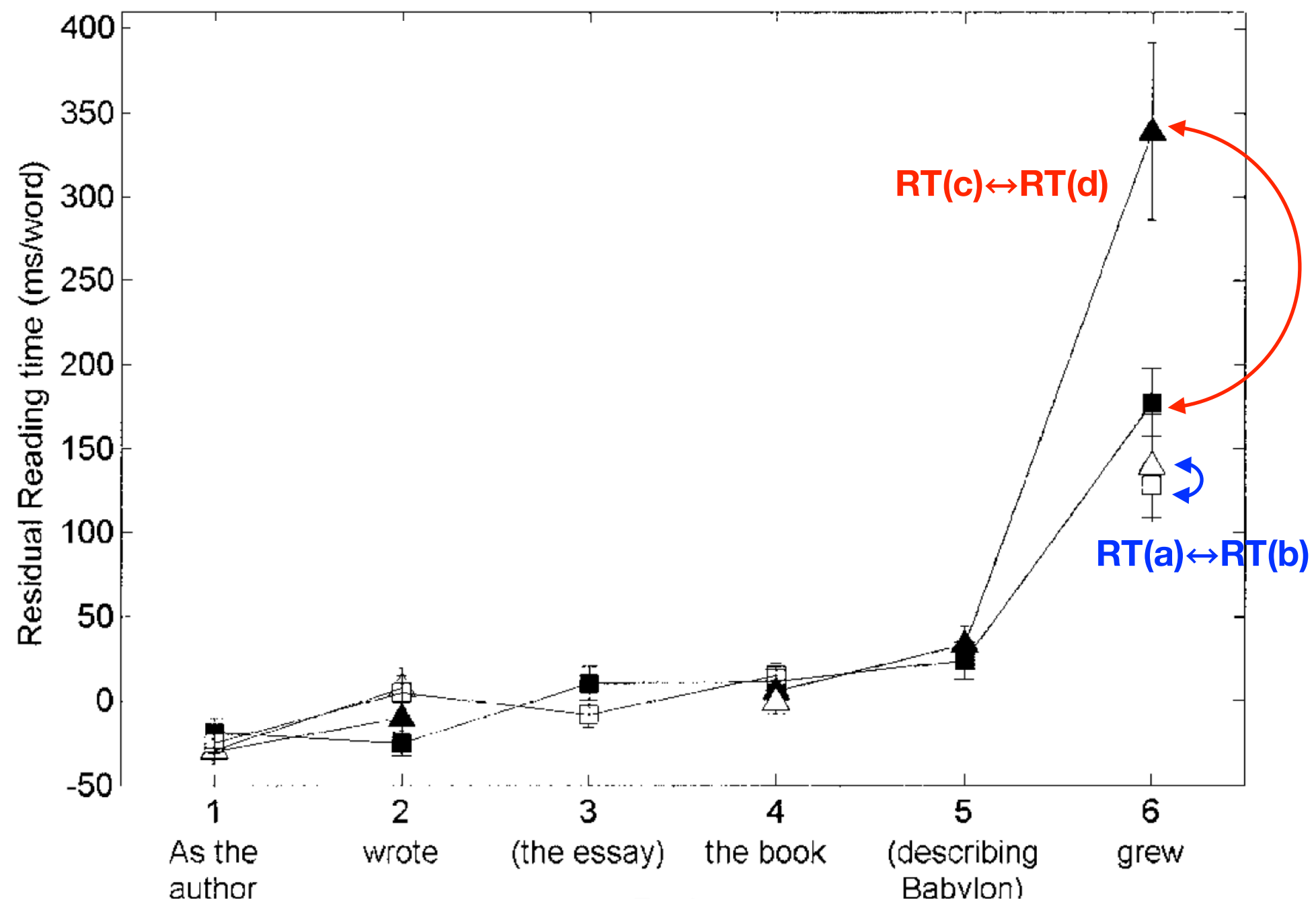
As the author wrote the essay the book **grew**.

▲(c) [long, -object]

As the author wrote the book describing Babylon **grew**.

■(d) [long, +object]

As the author wrote the essay the book describing Babylon **grew**.



"Digging in" in human NP/Z garden-pathing

△(a) [short, -object]

As the author wrote the book **grew**.

□(b) [short, +object]

As the author wrote the essay the book **grew**.

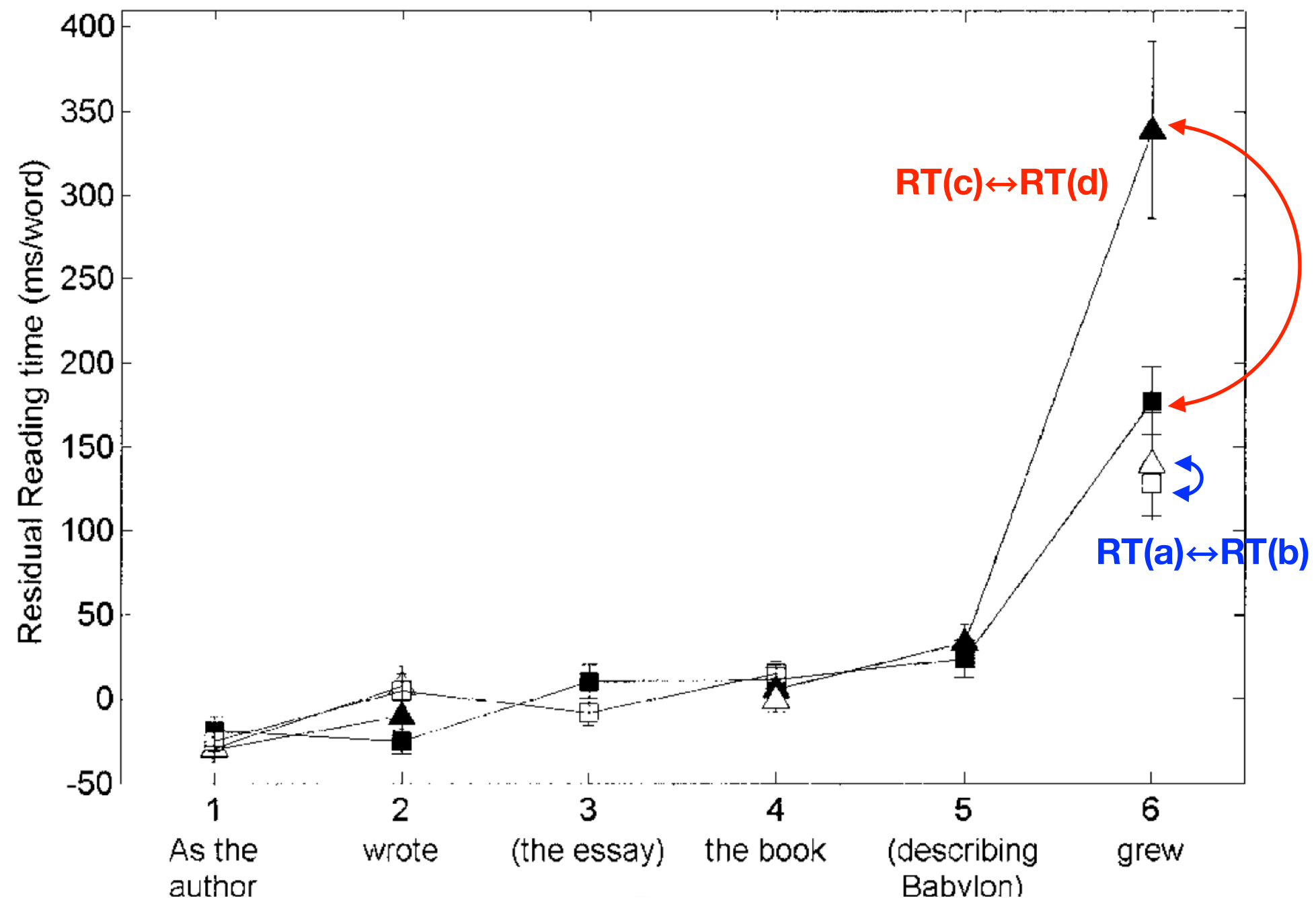
▲(c) [long, -object]

As the author wrote the book describing Babylon **grew**.

■(d) [long, +object]

As the author wrote the essay the book describing Babylon **grew**.

Surprisal in neural
language models
doesn't capture the
human "digging-in"
effect



NP/Z garden pathing: summary

- All models show evidence of a syntactic garden path that can be blocked by a comma
- Only models with larger amounts of data show verb transitivity-based garden-path modulation
- Not all models robustly maintain syntactic state-like distinctions over long stretches of intervening material
 - Explicit grammatical representations seem to help with this

References

- Adams, B. C., Clifton, C., Jr., & Mitchell, D. C. (1998). Lexical guidance in sentence processing? *Psychonomic Bulletin & Review*, 5 (2), 265–270.
- Choe, D. K., & Charniak, E. (2016). Parsing as language modeling. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 2331–2336).
- Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 2015 meeting of the Association for Computational Linguistics* (pp. 334–343).
- Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent Neural Network Grammars. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14, 178–210.
- Futrell, R., Wilcox, E., Morita, T., & Levy, R. (2018). RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. Version 1. *arXiv: 1809.01329 [cs.CL]*
- Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., & Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 32–42).
- Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1195–1205). New Orleans, Louisiana: Association for Computational Linguistics.
- Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.
- Kuncoro, A., Ballesteros, M., Kong, L., Dyer, C., Neubig, G., & Smith, N. A. (2017). What do Recurrent Neural Network Grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.
- Mitchell, D. C. (1987). Lexical guidance in human parsing: Locus and processing characteristics. In M. Coltheart (Ed.), *Attention and performance xii: The psychology of reading*. London: Erlbaum.
- Staub, A. (2007). The parser doesn't ignore intransitivity, after all. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 33 (3), 550-569.
- Stern, M., Fried, D., & Klein, D. (2017). Effective inference for generative neural parsing. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1695–1700).
- Tabor, W., & Hutchins, S. (2004). Evidence for self-organized sentence processing: Digging in effects. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 30 (2), 431–450.
- van Gompel, R. P. G., Pickering, M. J., & Traxler, M. J. (2001). Reanalysis in sentence processing: Evidence against current constraint-based and two-stage models. *Journal of Memory and Language*, 45, 225–258.