Combinatory Categorial Grammars

Lexicalized Semantically Guided Syntax

Yonatan Bisk

The People (a very abridged version)

Theory

Mark Steedman (Edinburgh)

Data and Parsers

Jason Baldridge (Google)

Julia Hockenmaier (UIUC)

Weakly Supervised Parsing

Me!

Dan Garrette (Google)

Efficient Parsing

Stephen Clark (Cambridge)

James Curran (Sydney)

Mike Lewis (FAIR)

Mirella Lapata
Adam Lopez
Johan Boss
Cem Bozsahin
Michael White
…

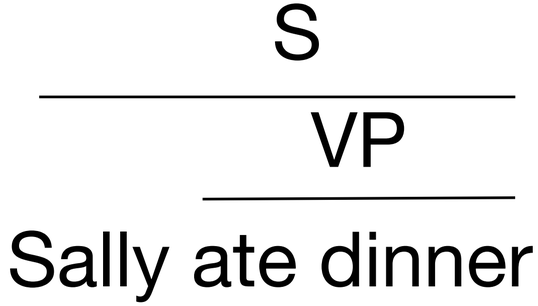
Semantic Parsing

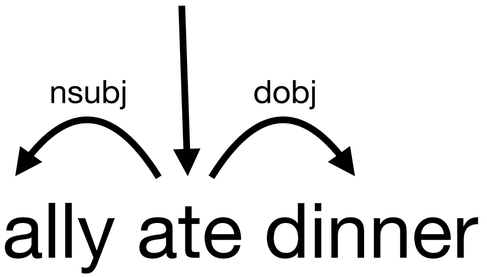
Luke Zettlemoyer (UW)

Yoav Artzi (Cornell)

Siva Reddy (McGill)

Where is syntax?

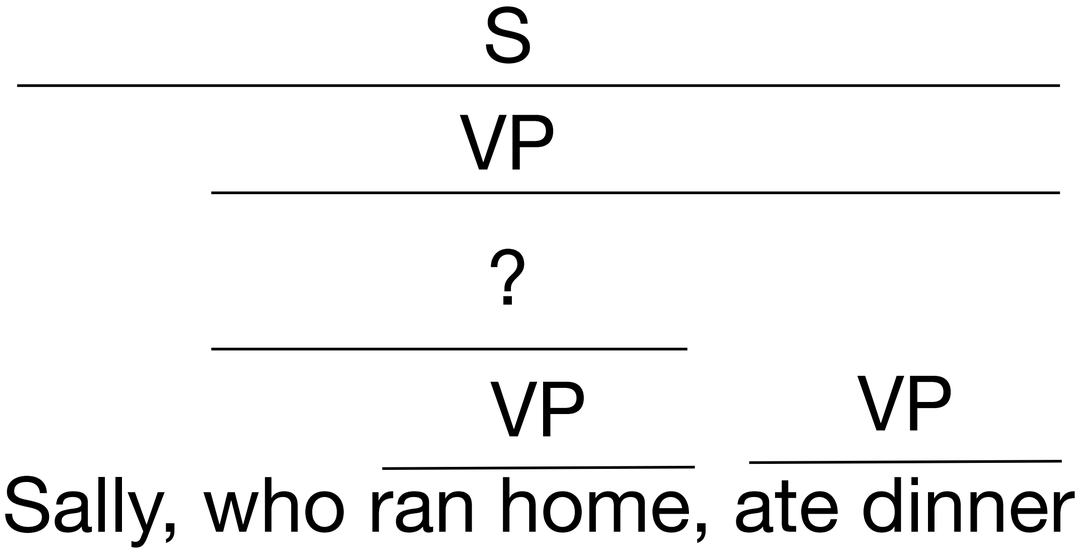


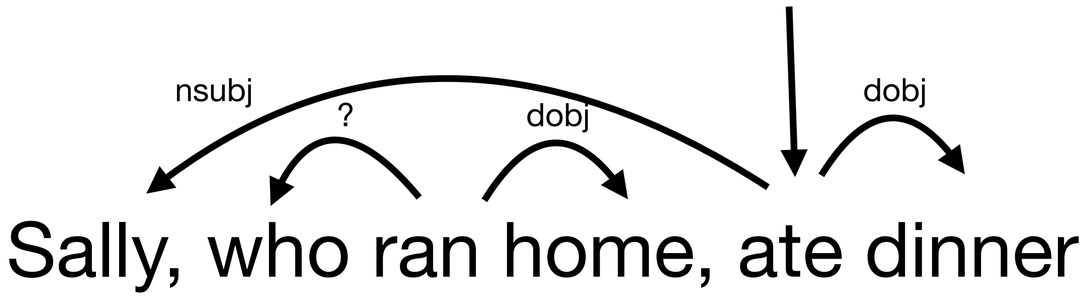


- Hierarchical syntactic labels
- Where did the labels come from?
- Are all VPs the same? …

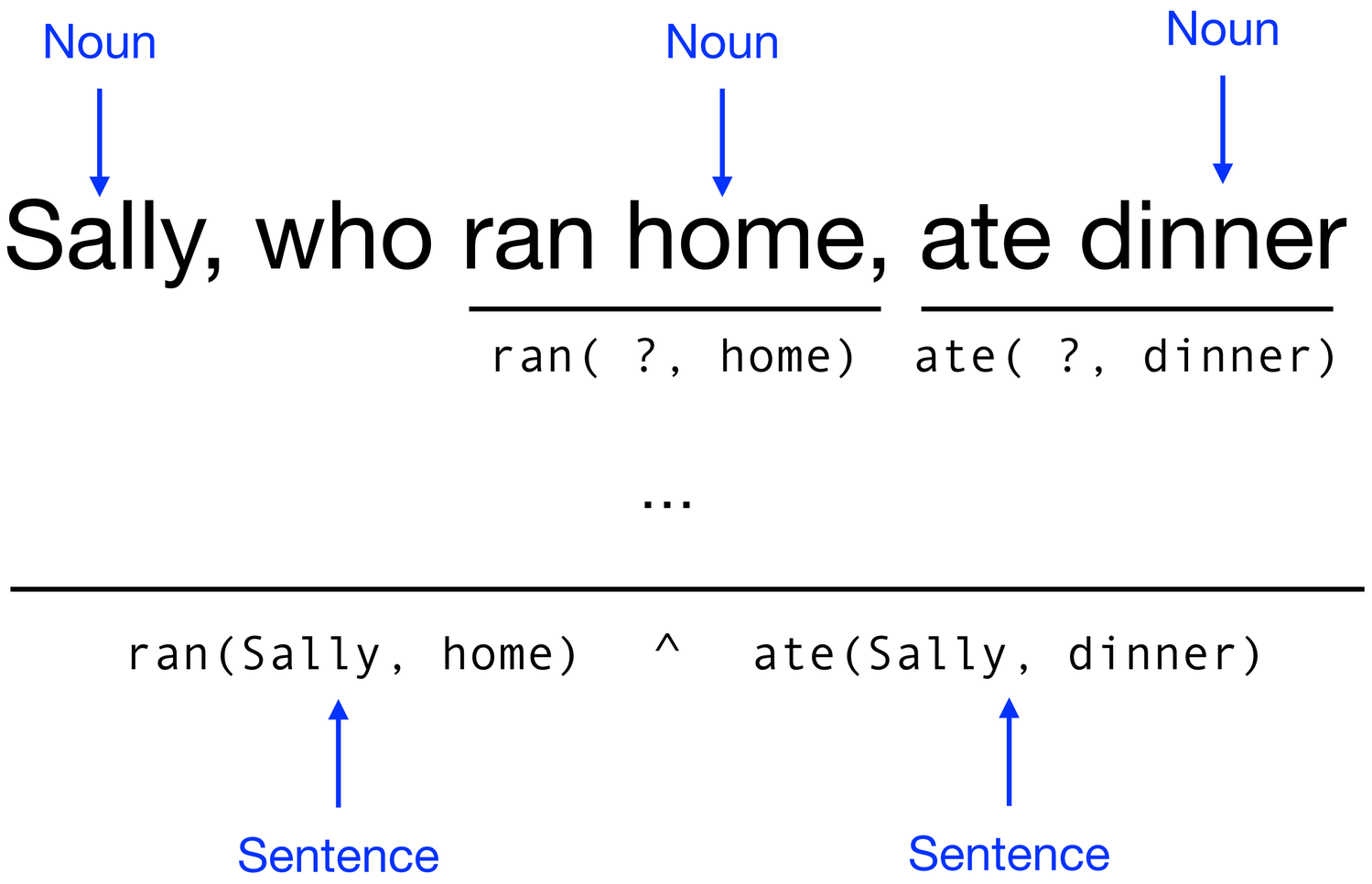
- Semantic labels
- How do you represent long-distance effect?
- Are all languages really tree-structured?

Where is syntax?

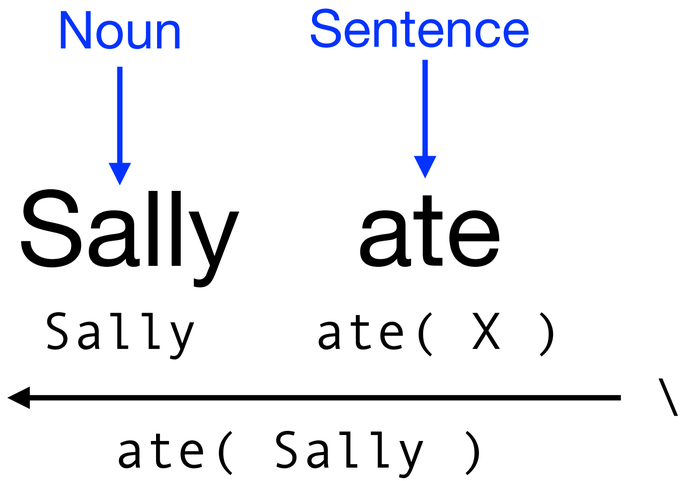




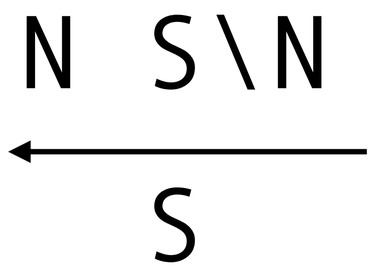
Where is syntax?



Syntax as Functions



Function Application



`S\N` is a function that
if applied to a `N` on the left
returns an `S`

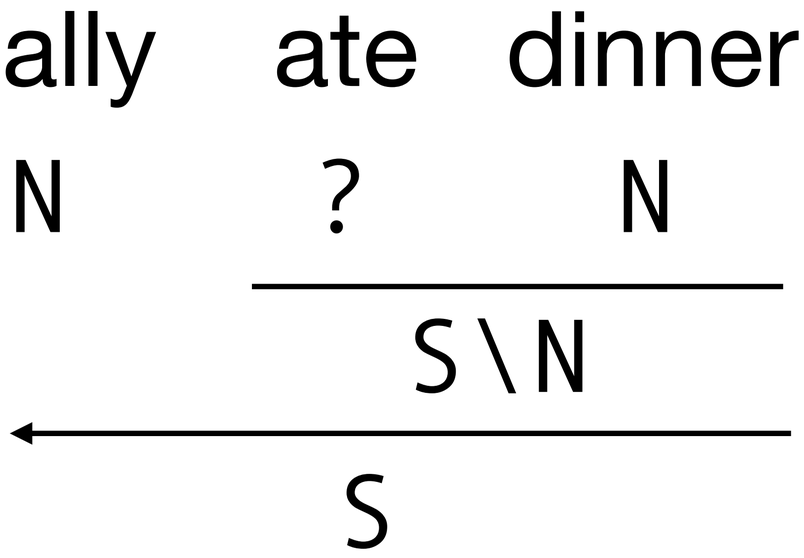
Grammar:

Sally: `N`

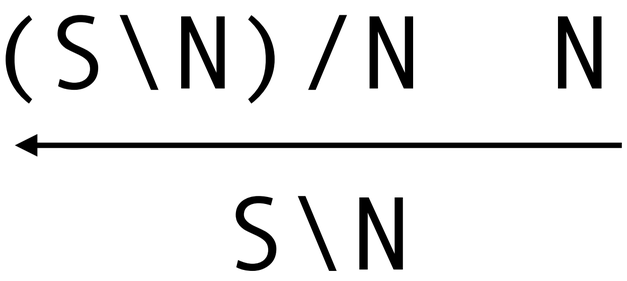
ate: `S\N`

Every word is a function
or an argument

Syntax as Functions



Functions returning functions



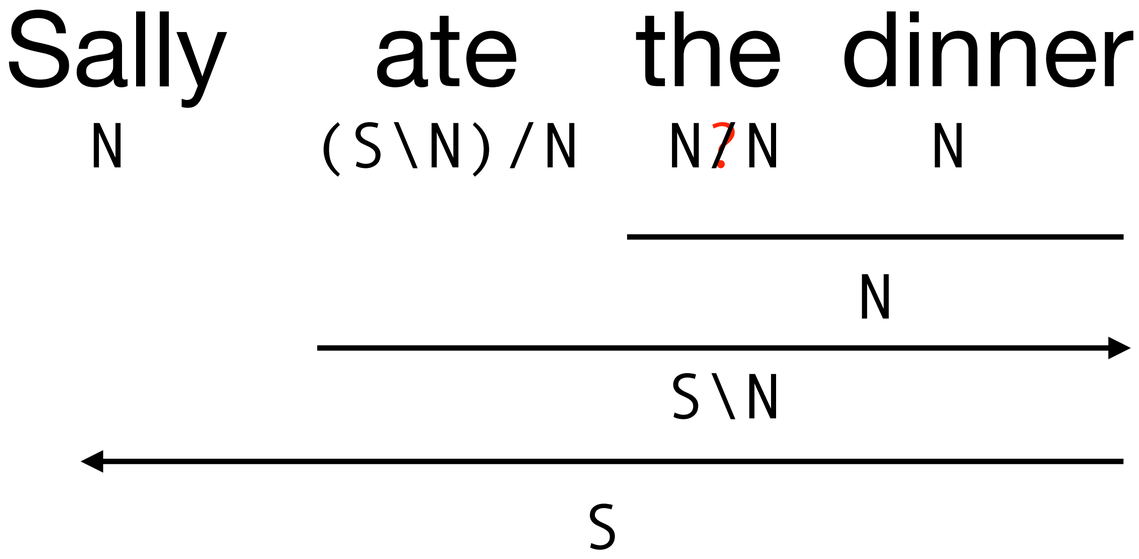
Grammar:

Sally: N

dinner: N

ate: S\N, (S\N)/N

Syntax as Functions



?: Takes in N (on right) and returns N

N/N

Grammar:

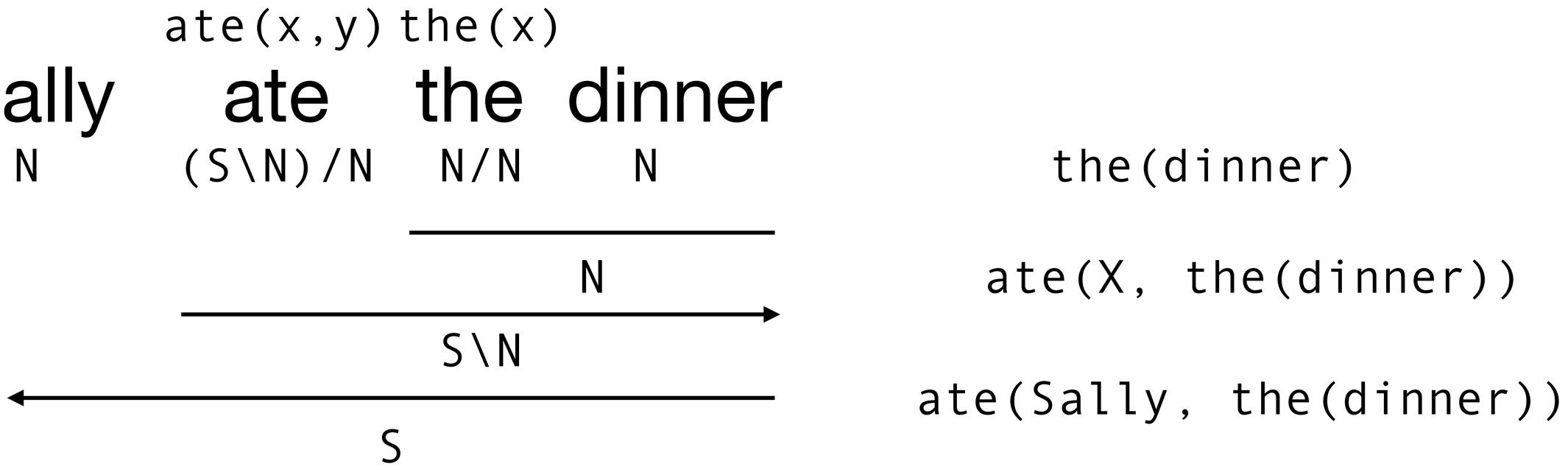
Sally: N

dinner: N

ate: S\N, (S\N)/N

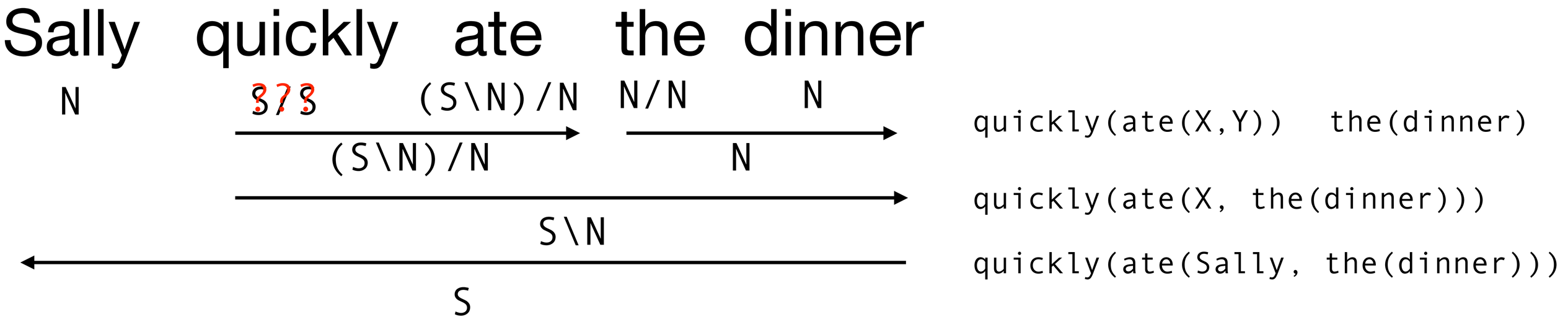
the: N/N

Syntax as Semantics



Function Composition

S/S (S\N)/N ⟶ (S\N)/N



Recap

Grammar

Sally	N
dinner	N
the	N/N
ate	S\N, (S\N)/N
quickly	S/S

Function Math

Application

$$X/Y \quad Y \rightarrow X$$

$$Y \quad X\backslash Y \rightarrow X$$

Composition

$$X/Y \quad Y/Z \rightarrow X/Z$$

$$X/Y \quad Y\backslash Z \rightarrow X\backslash Z$$

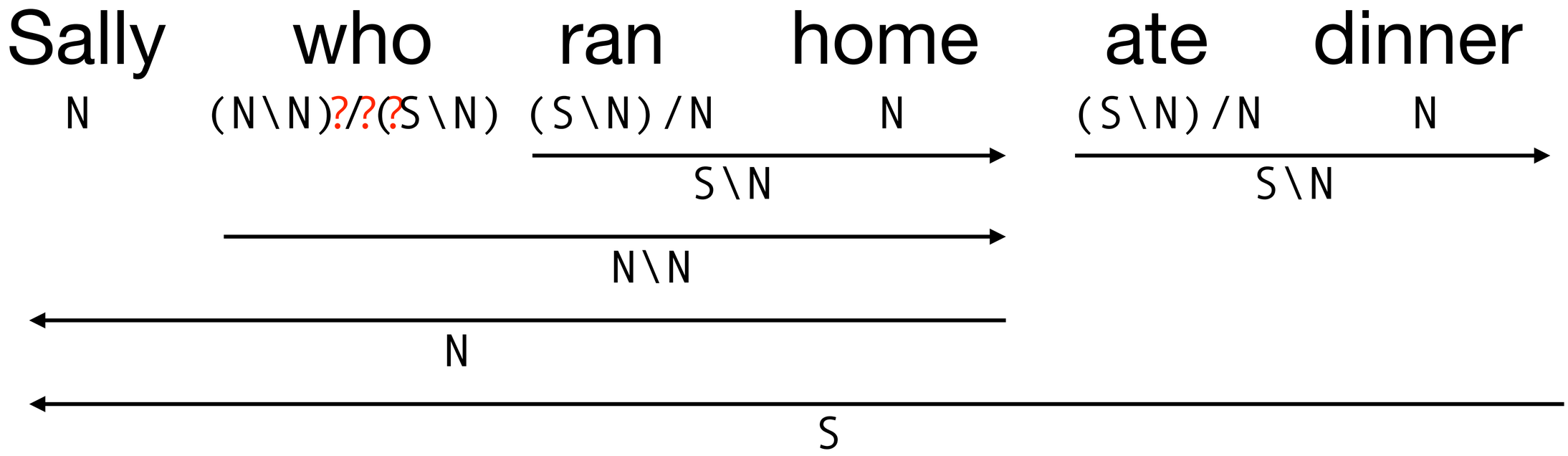
$$Y/Z \quad X\backslash Y \rightarrow X/Z$$

$$Y\backslash Z \quad X\backslash Y \rightarrow X\backslash Z$$

Puzzle Time

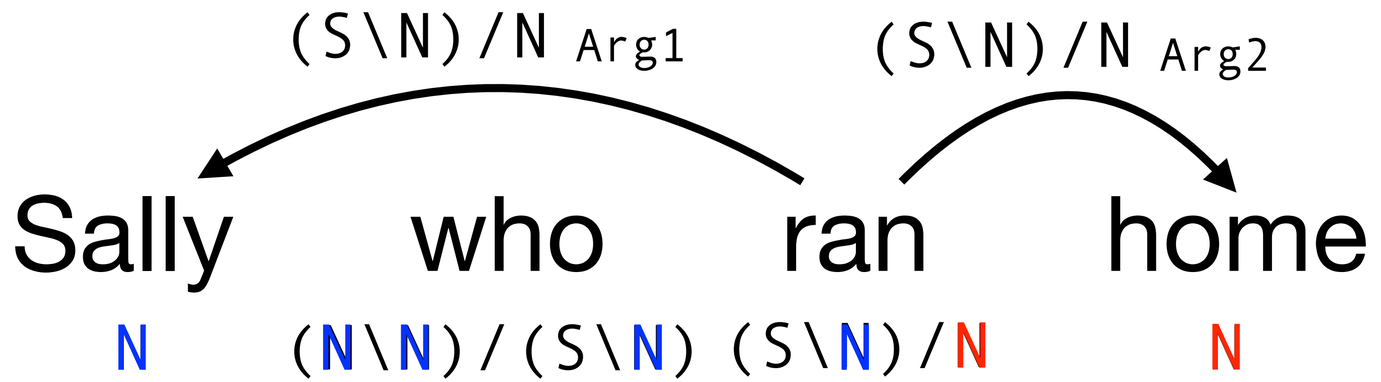
What is a relative clause?

A noun modifier



`ran(Sally, home)` `ate(Sally, dinner)`

Unification & Dependencies

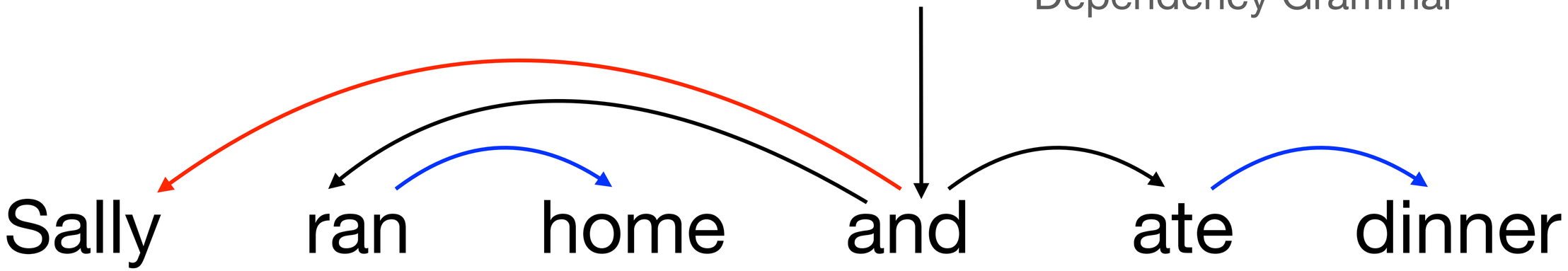


`ran(Sally, home)`

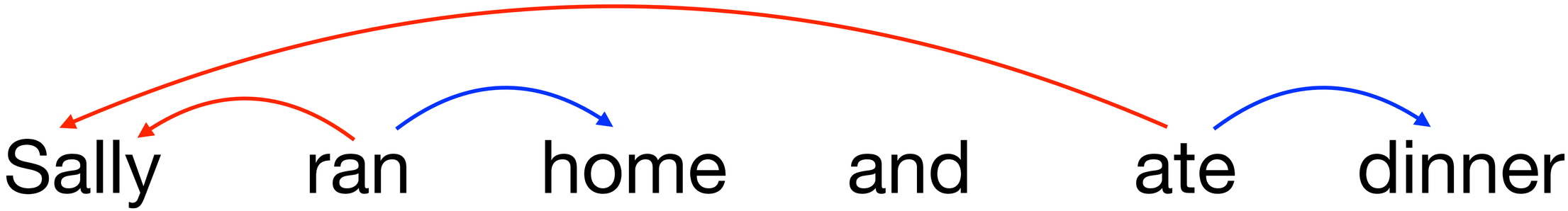
Note: These are more fine-grained labels than nsubj/dobj

Coordination

Dependency Grammar

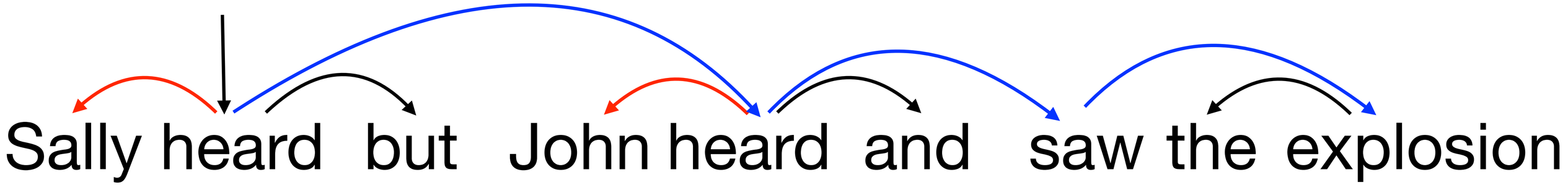


CCG Dependency Grammar

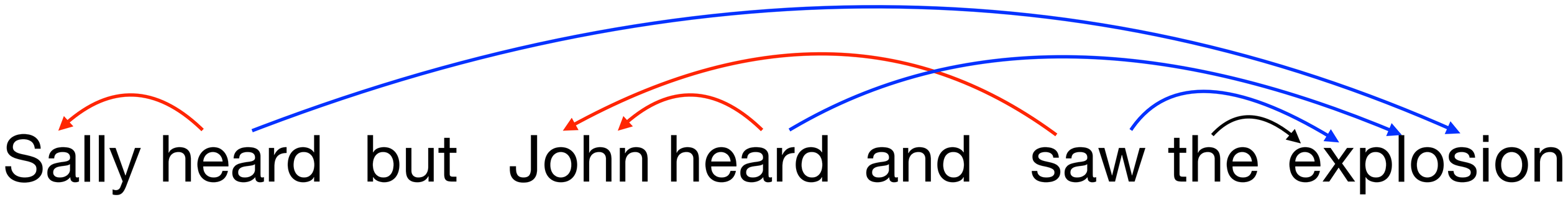


Coordination

Dependency Grammar



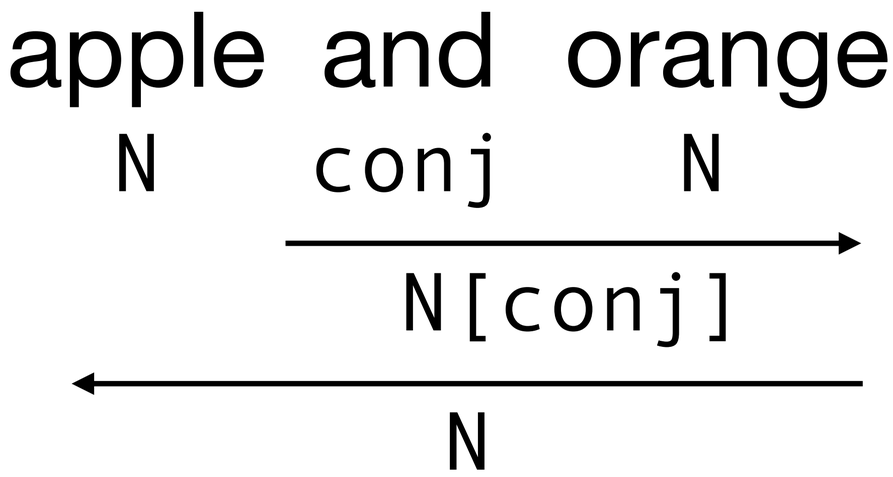
CCG Dependency Grammar

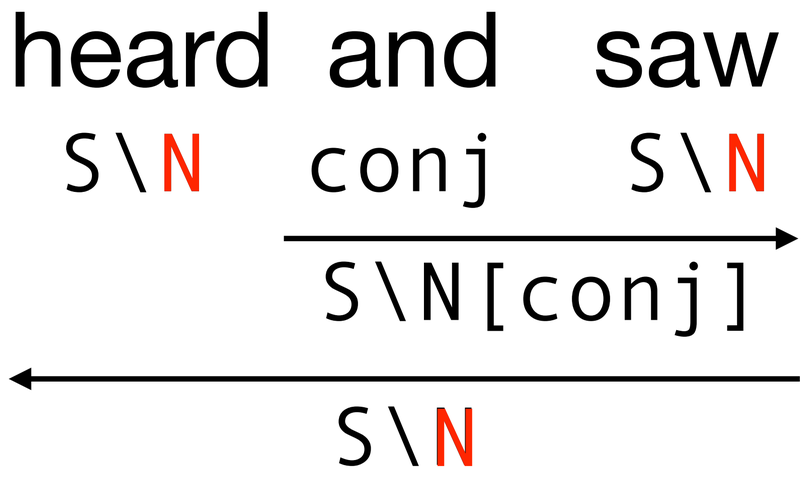


Coordination

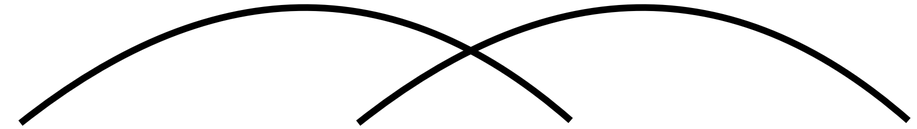
$$X \quad conj \longrightarrow X[conj]$$

$$conj \; X[conj] \longrightarrow X$$

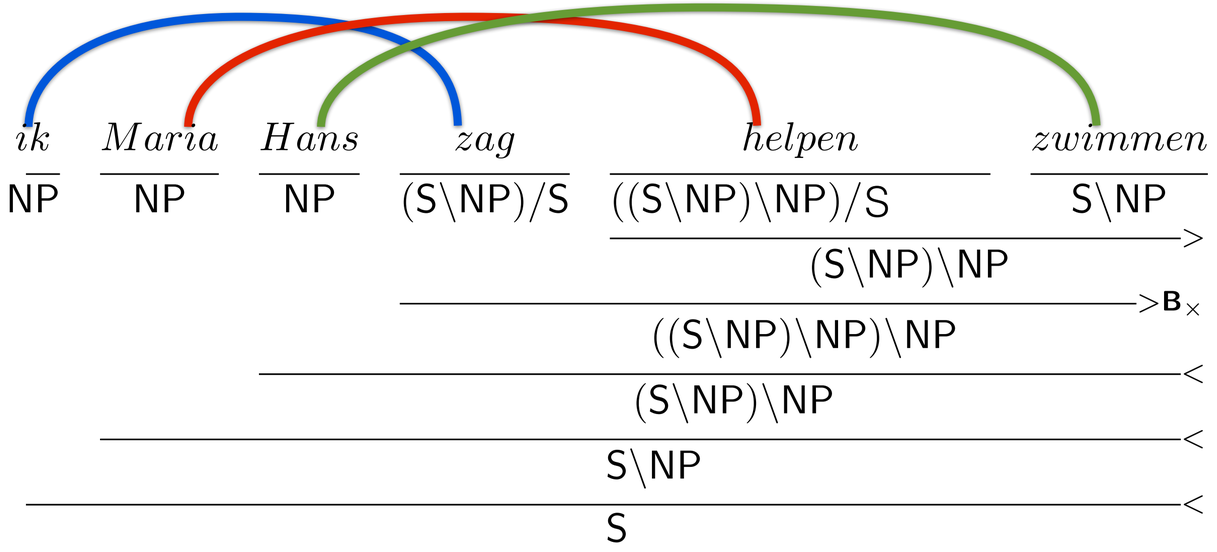




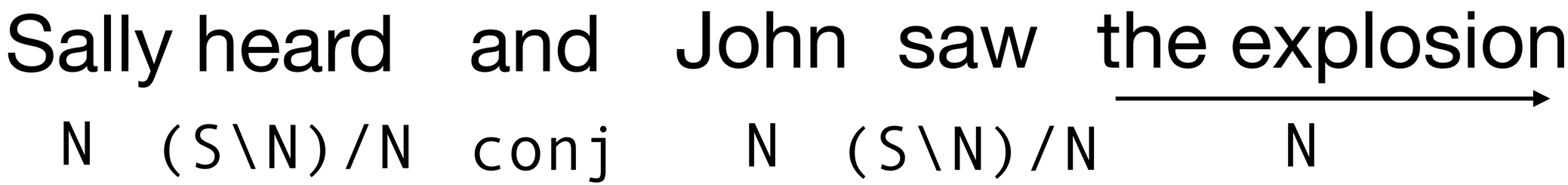
Side note: Crossing Dependencies Are Real



I ate the red and yellow, apple and banana, respectively

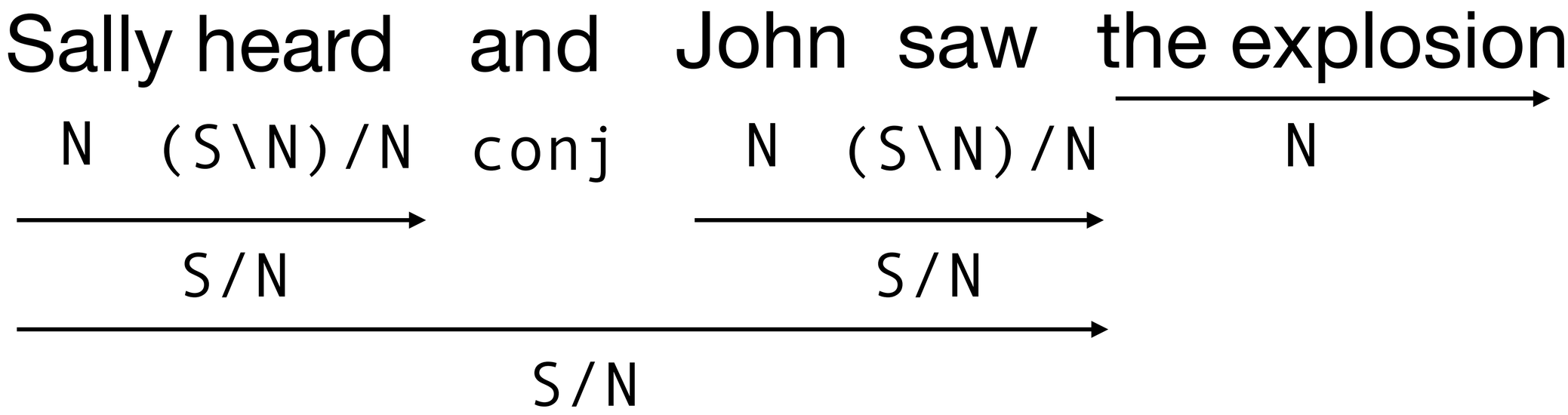


Puzzle Time



What should apply to what?

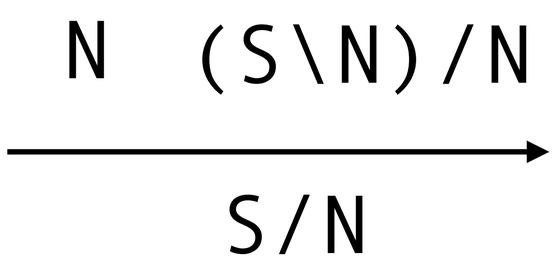
Puzzle Time


Sally heard and John saw the explosion
N (S\N)/N conj N (S\N)/N N
S/N S/N
S/N

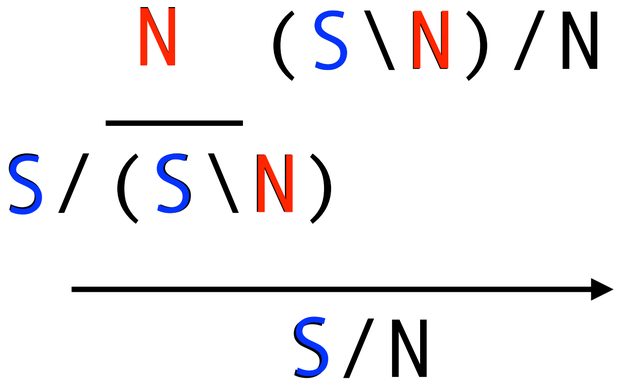
Puzzle Time

Fill Arg 1 before Arg 2

Sally heard



Sally heard



Type-Raising

Lexicon & Rules

Grammar

Sally	N
dinner	N
the	N/N
ate	S\N, (S\N)/N
quickly	S/S
and	conj

Application

$$X/Y \quad Y \rightarrow X$$
$$Y \quad X\backslash Y \rightarrow X$$

Composition

$$X/Y \quad Y/Z \rightarrow X/Z$$
$$X/Y \quad Y\backslash Z \rightarrow X\backslash Z$$
$$Y/Z \quad X\backslash Y \rightarrow X/Z$$
$$Y\backslash Z \quad X\backslash Y \rightarrow X\backslash Z$$

Conjunction

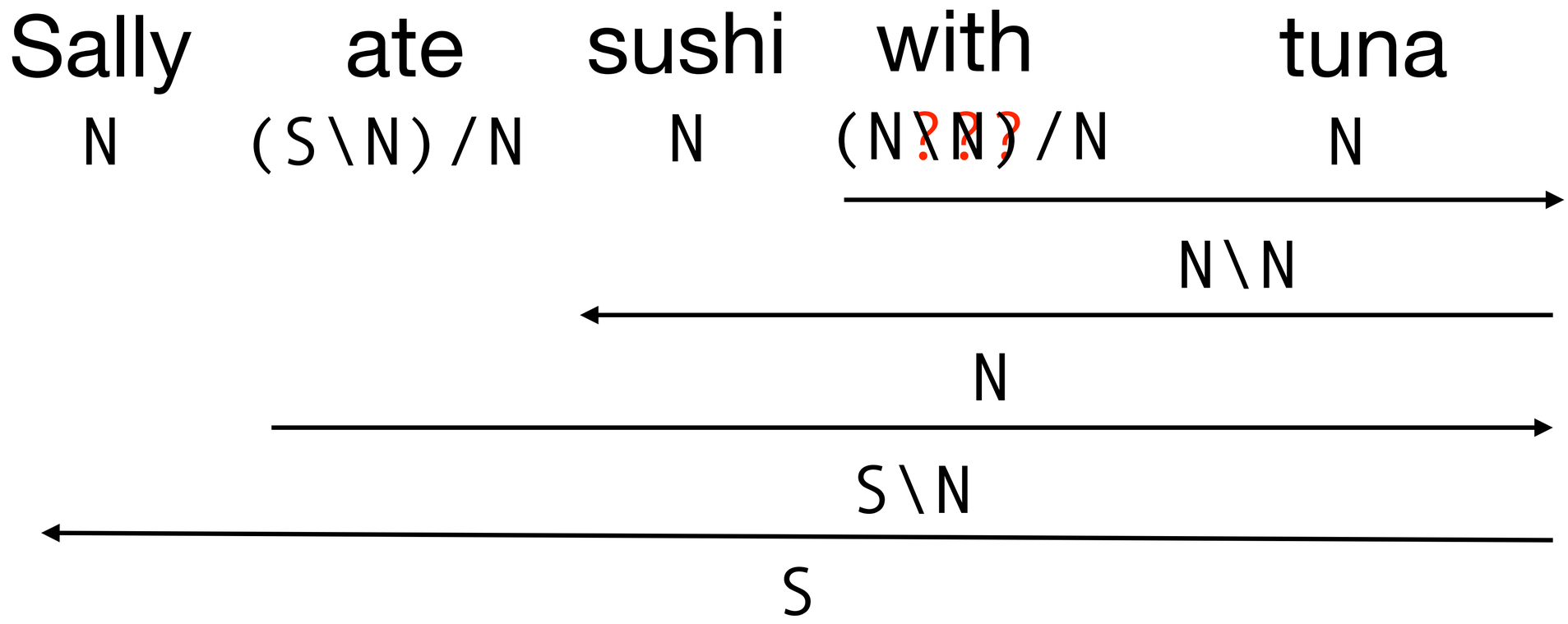
$$X \quad conj \rightarrow X[conj]$$
$$X[conj] \quad X \rightarrow X$$

Type-Raising

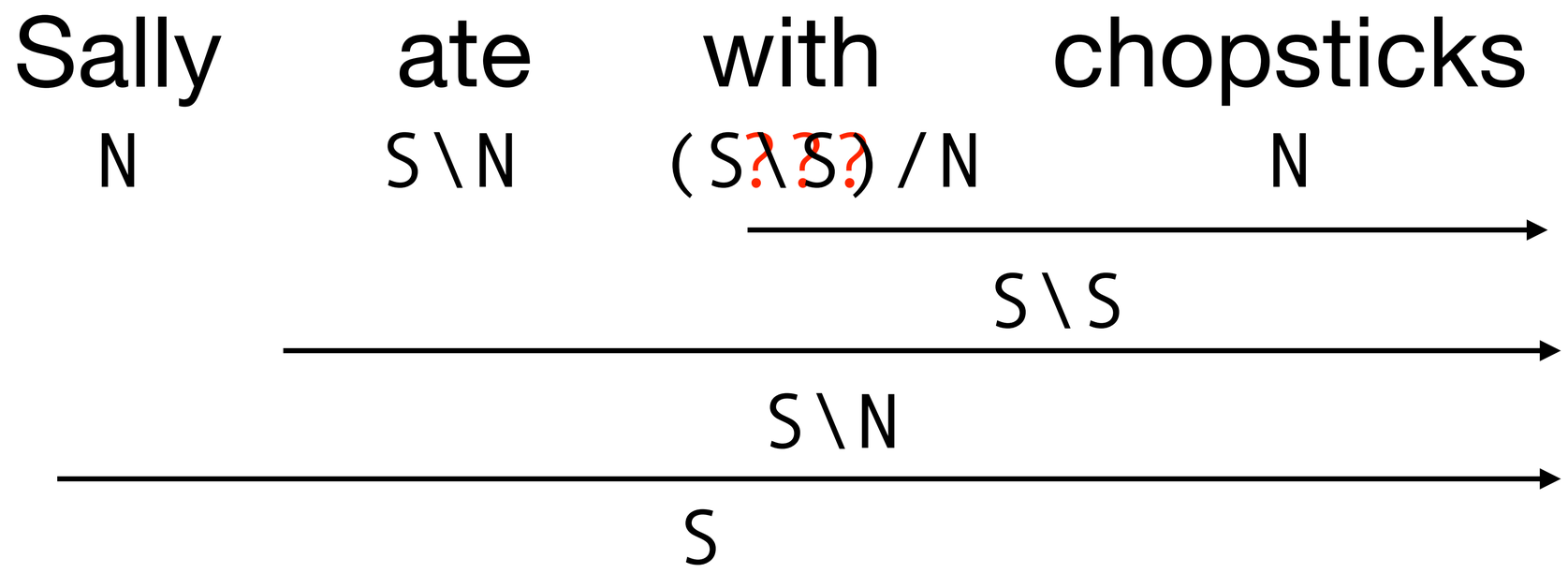
$$N \rightarrow S/(S\backslash N)$$
$$N \rightarrow S\backslash(S/N)$$

That’s it!
Just make up categories

Puzzle Time



Puzzle Time



Sweeping things under the rug

- It's really just `S`, `N`, and `conj`? Well… no

  NP, PP

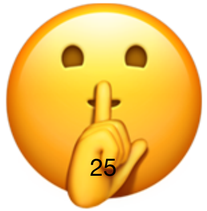
- Ok, so 5 categories? Sorta…

  S[adj], S[dcl], S[b],… NP[nb], …

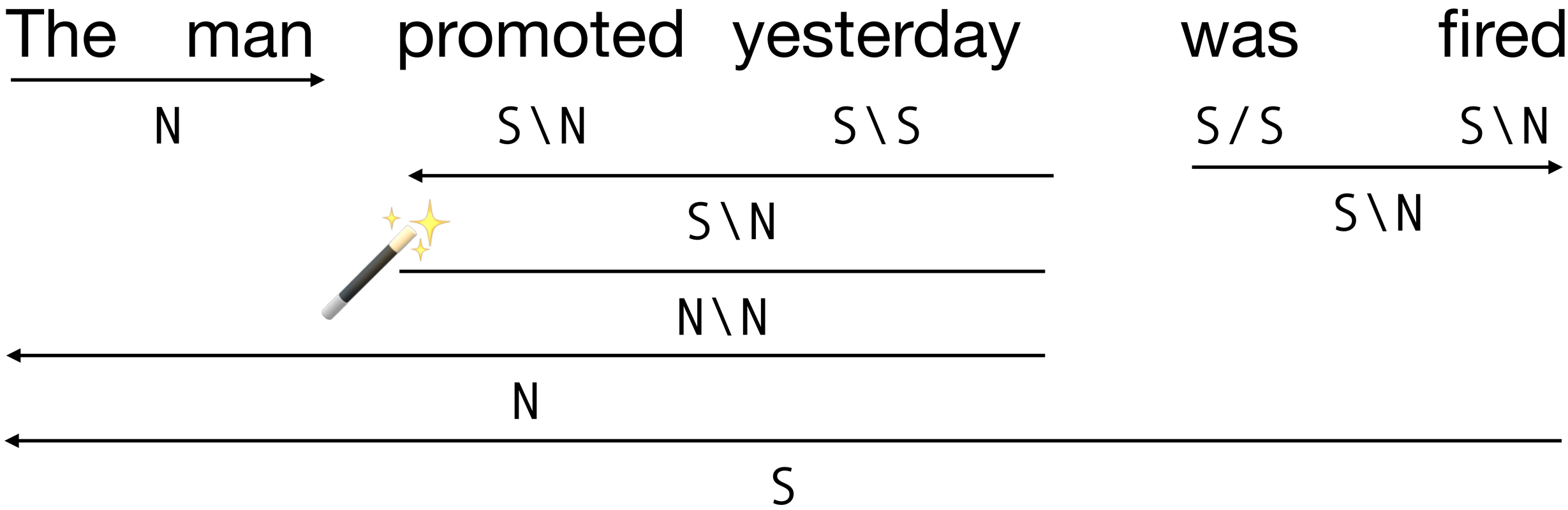
- OK, but we did learn all the rules right?

  Ugh,… "Yes" — ignore Type-Changing, it's not really real…

- Are you lying to make this formalism sound prettier then it really is?



The dirty …



As I said, everything is perfect, there are no questions, everyone is happy with this result.
CCG is beautiful and perfect.

Why CCG?

Could be SQL, SPARQL, python, etc

Sally, who ran home, ate dinner

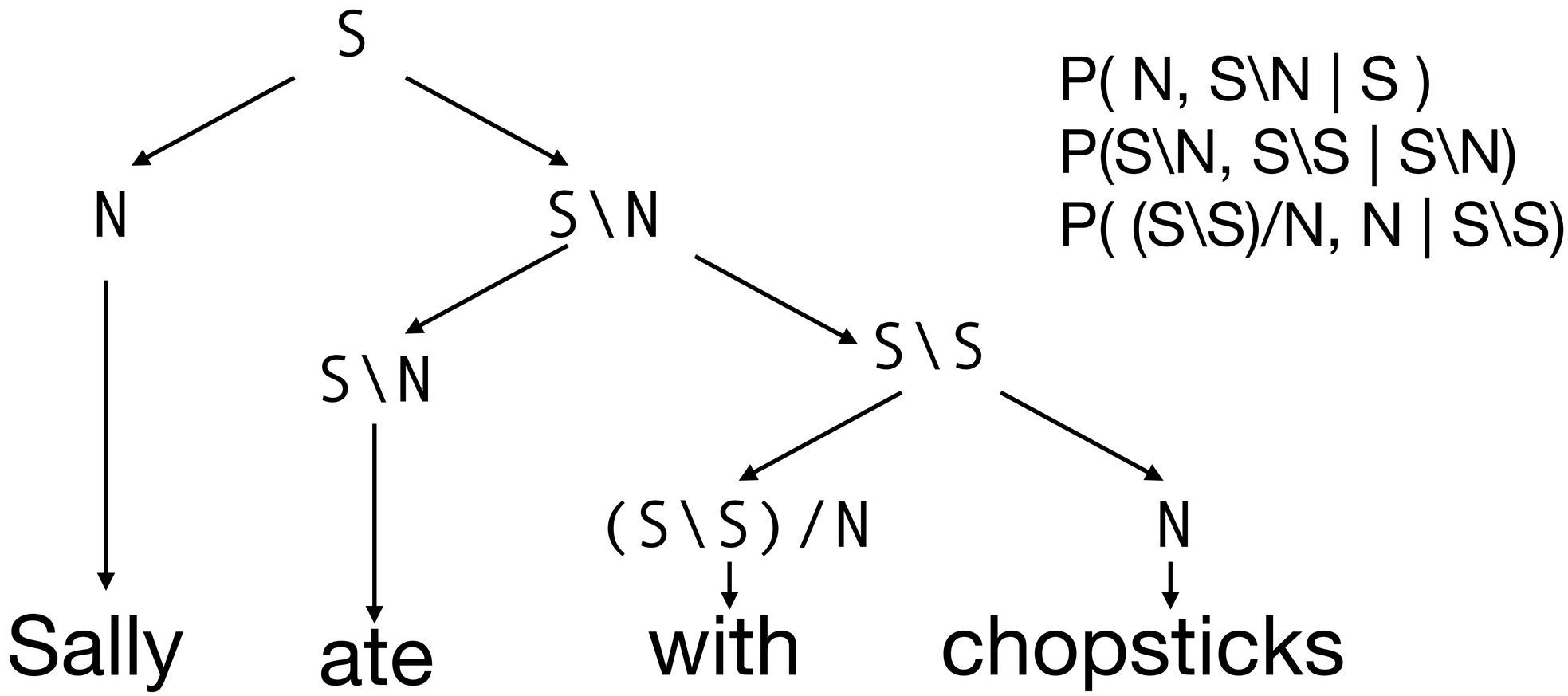
`λy.λx.f(x, y)`

ran	`(S\N)/N`	`λy.λx.run(x, y)`
ate	`(S\N)/N`	`λy.λx.eat(x, y)`
who	`(N\N)/(S\N)`	`λf.f`

`ran(Sally, home)` `ate(Sally, dinner)`

Modeling

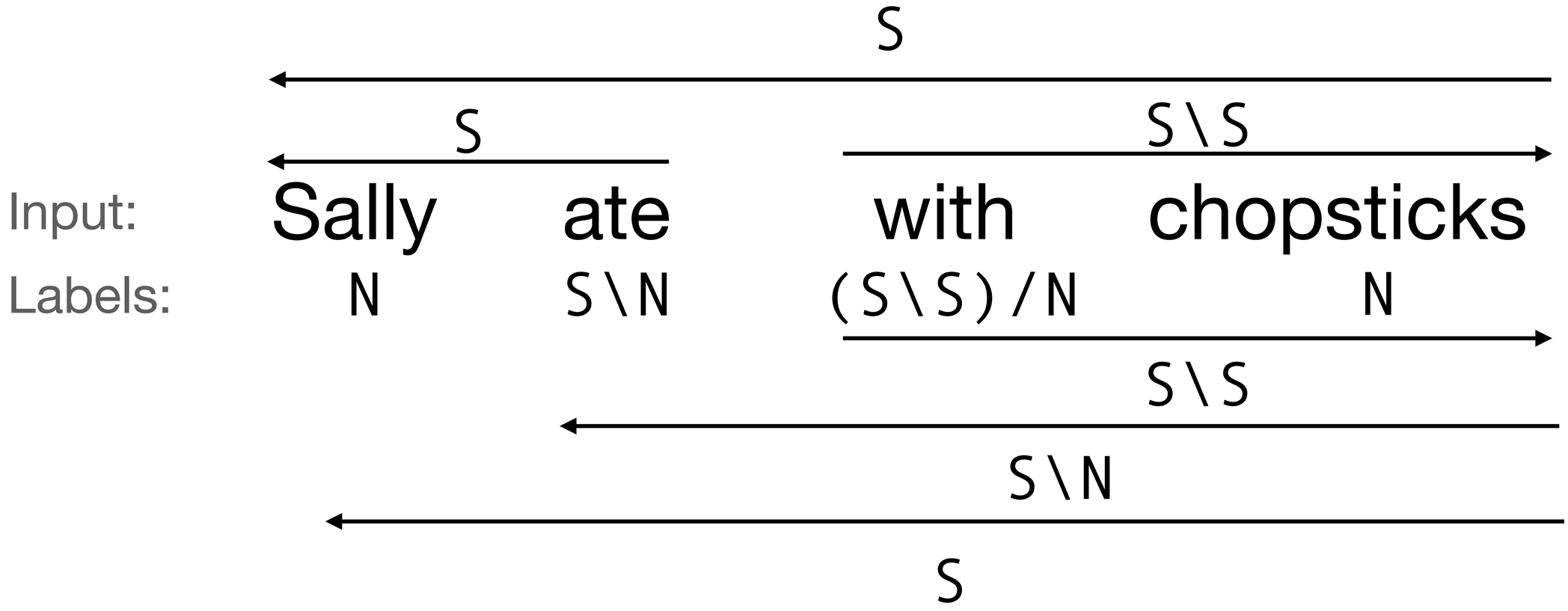
How should we define a probabilistic model?



P(N, S\N | S)
P(S\N, S\S | S\N)
P((S\S)/N, N | S\S)

Modeling

Supertagging



Normal-form parsing for Combinatory Categorial Grammars with generalized composition and type-raising — Hockenmaier 2010

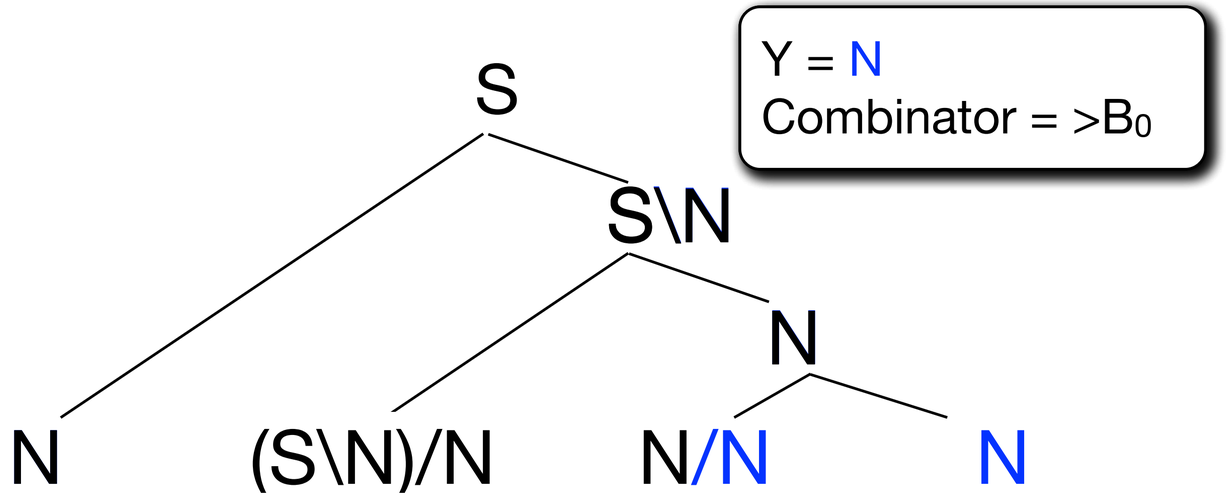
A* CCG Parsing with a Supertag-factored Model Lewis 2014 — http://4.easy-ccg.appspot.com/do_parse?sentence=Fruit+flies+like+a+banana&nbest=5

Supertag LSTM Analysis

				LSTM			
Supertag	F-For	Forward	Backward	bi–LSTM	+LM(g–train)	ss–train–1	ss–train–5
(NP\NP)/NP	90.00	88.89	81.91	92.09	92.18	91.72	**92.31**
((S\NP)\(S\NP))/NP	75.75	69.53	61.60	**80.38**	78.21	79.91	78.77
S[dcl]\NP	77.29	61.14	58.52	**84.28**	83.41	82.97	80.35
(S[dcl]\NP)/NP	91.39	56.58	69.86	92.34	92.46	92.46	**92.82**
((S[dcl]\NP)/PP)/NP	42.30	30.77	42.31	56.41	**64.10**	62.82	60.26
(S[dcl]\NP)/(S[adj]\NP)	86.80	22.84	83.25	87.31	**88.83**	87.82	86.80
((S[dcl]\NP)/(S[to]\NP))/NP	86.49	56.76	75.68	**94.59**	91.89	91.89	91.89

Supertagging with LSTMs Vaswani 2016

Modeling the Arguments



Induced Lexicons: Adjectives

English

Big	Ball
N/N	N

Adj Obj

Arabic

كرة	كبيرة
N	**N\N**
(ball)	(big)

Obj Adj

Induced Lexicons: Verbs

English

S V O

The man	wrote	a letter
N	**(S\N)/N**	N

Child Directed Speech

∅ V O

∅	write	a letter
	S/N	N

Arabic

V S O

كتب	الرجال	رسالة
(S/N)/N	N	N
(wrote)	(the man)	(a letter)

Induced Lexicons: Adpositions

English

ran	on	beach
S\N	**(S\S)/N**	N

V ADP O

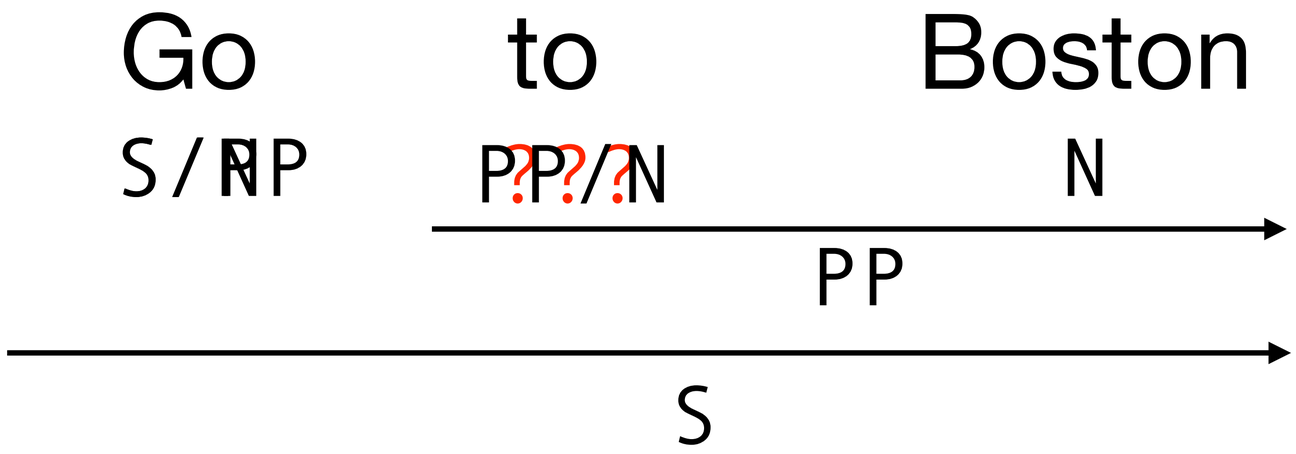
Japanese

浜	を	走った
N	**(S/S)\N**	S\N
(beach)	(on)	(ran)

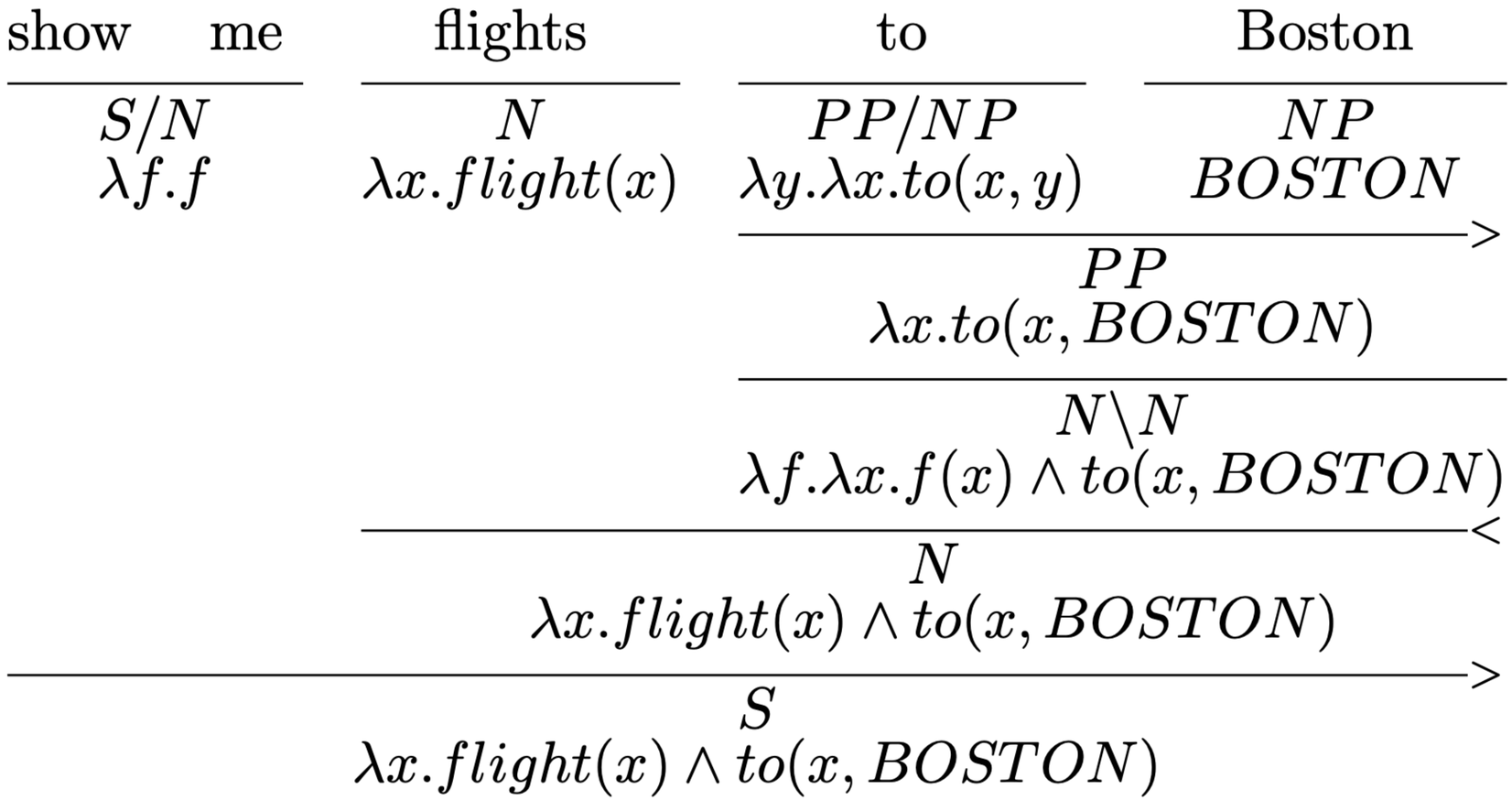
O ADP V

Prepositions can be tricky

Is “to Boston” a modifier?

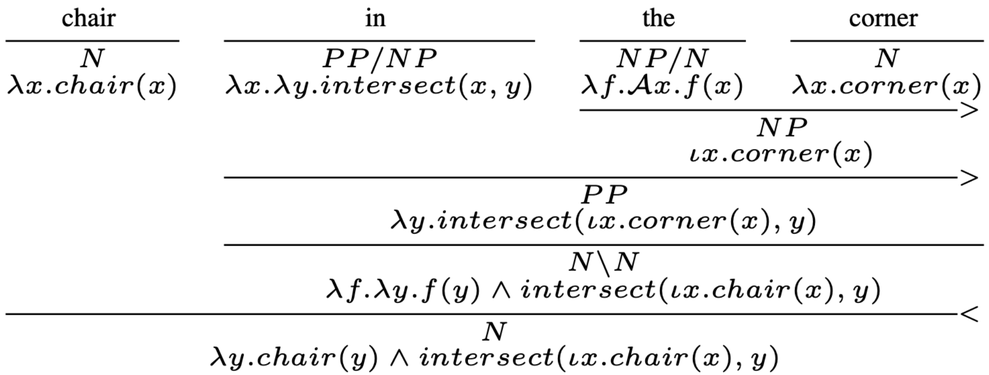


Stolen from Artzi ACL 2013



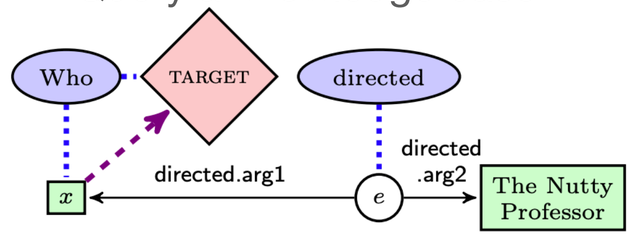
You parse so you can do something

How does a robot check if it's at the right location?



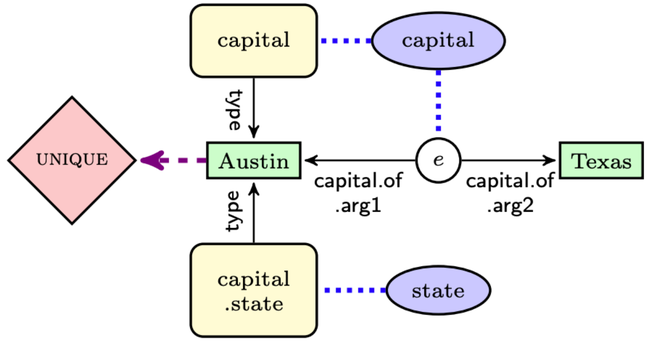
Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions — Artzi 2013

Query a knowledge base



TARGET(x) ∧ directed.arg1(e, x) ∧
directed.arg2(e, TheNuttyProfessor)

(a) Who directed The Nutty Professor?



UNIQUE(Austin) ∧ capital(Austin) ∧ capital.state(Austin) ∧
capital.of.arg1(e, Austin) ∧ capital.of.arg2(e, Texas)

(b) Austin is the state capital of Texas.

Large-scale Semantic Parsing without Question-Answer Pairs — Reddy 2014

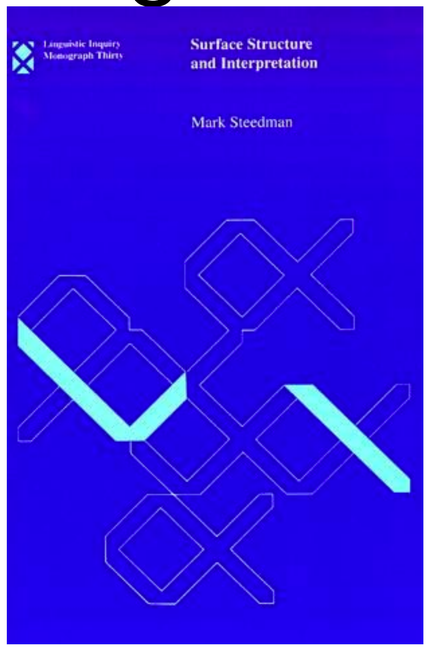
Where to learn more?

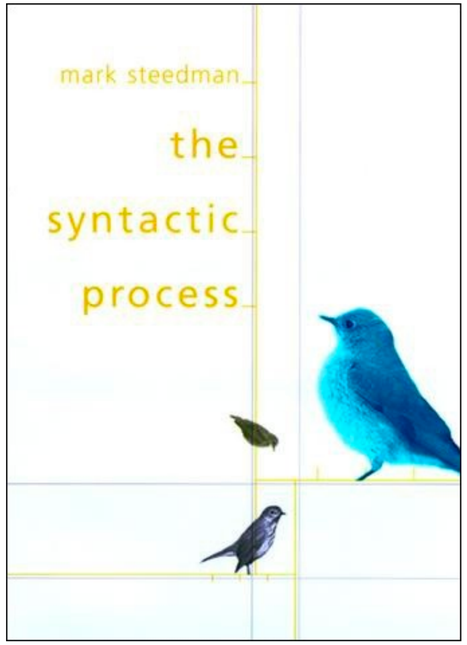
Semantic Parsing and Modeling

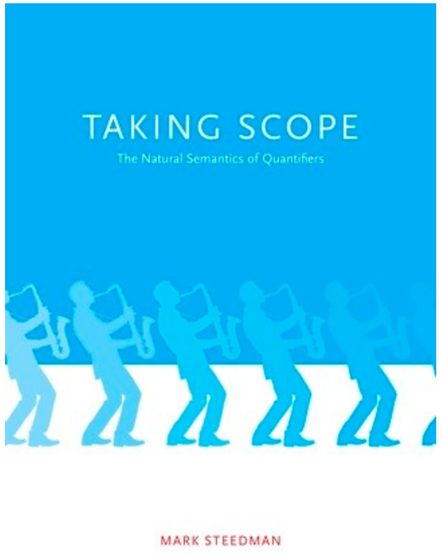
https://yoavartzi.com/tutorial/

Linguistics

Mark Steedman







Jazz

http://jazzparser.granroth-wilding.co.uk/Parser.html