Prescriptive versus Descriptive

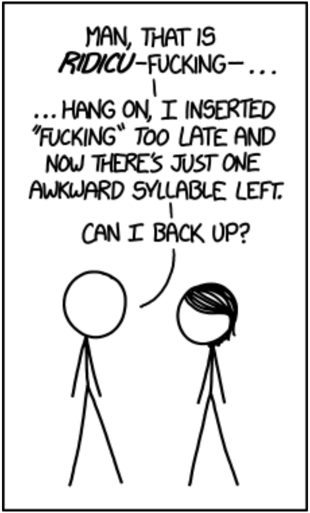
- Prescriptive (largely proscriptive): old-school grammar; mostly bogus
  - Don't end a sentence with a preposition
  - Don't split an infinitive: to boldly go
  - Avoid the passive voice
  - Don't use double negatives
- Double negatives in Polish (Bender, Sag, Wasow's example)

Marysia	niczego	nie	dala	Jankowi
Mary	nothing	not	gave	John

  Mary did not give John anything
- Descriptive: what people actually speak or write
  - Does anything go?
- For your own professional writing, follow the prescriptions!

XKCD on Expletive Infixation

An illustration of descriptive grammar



Where would you place it?
— ri — di — cu — lous —

Subtle Constraints in Descriptive Grammar

How do we explain these examples? (* indicates unacceptability)

- Bender, Sag, Wasow's examples
  - F— yourself!
  - Go f— yourself!
  - F— you!
  - *Go f— you!
- Wanna contraction (from Wikipedia)
  - Who does Vicky want to vote for?
    $\Rightarrow$ Who does Vicky wanna vote for?
  - Who does Vicky want to win?
    $\Rightarrow$ *Who does Vicky wanna win
- Gonna contraction
  - I am gonna get lunch
  - *I am gonna New York
- Gonna and wanna function like AUX verbs

Competence versus Performance

Chomsky's distinction

- Frederic Saussure
  - Langue: collective knowledge of language
  - Parole: what is observable
- Competence
  - Knowledge of language
  - What *native speakers* understand (abstract, ideal)
    - Standard of acceptability that is not prescriptive
  - Encoded in universal features or settings of universal parameters
- Performance
  - How the knowledge of language is used
  - How native speakers behave (concrete, noisy)

Constituency Structure

Constituent: set of words behaving as a single unit

- Phrase
- Theoretically established as
  - Having contiguous words
  - Nonoverlapping unless one phrase is entirely within another
- Appear in similar syntactic contexts, e.g., before or after a verb or a noun
  - But generally not the individual words within the phrase
  - Coordination: "X and Y" indicates X and Y have the same type
- Movable as a unit, e.g., *preposed* or *postposed*
  - But generally not the individual words within the phrase

I can write a letter	A letter is what I can write
I can write a long letter	A long letter is what I can write
*I can write a long	*A long is what I can write letter

Context-Free Grammar

In programming languages, we use parentheses

- Give examples of surrogates for parentheses in English

Context-Free Grammar

Part of the Chomsky hierarchy

- Stronger than a regular grammar
- Previous works assumed a regular grammar for human language
- Recall the pumping lemma
- Weaker than a context sensitive grammar
- CFGs are needed to handle natural structure in human languages: think of matching parentheses
- Bender, Sag, Wasow's example:
  - That Sandy left bothered me
  - That that Sandy left bothered me bothered Kim
  - That that that Sandy left bothered me bothered Kim bothered Bo
- A grammar describes (and generates) all and only the valid finite strings over a given alphabet
- For NL, the alphabet is words or tokens in a lexicon (Jurafsky seems to use "lexicon" oddly in this setting)

Formalizing a Context-Free Grammar

- Components of a grammar, $G = \langle N, \Sigma, R, S \rangle$
  - Σ, a finite alphabet or set of *terminal* symbols
  - N, a finite set of *nonterminal* symbols, $N \cap \Sigma = \emptyset$
  - $S \in N$, a *start* symbol (distinguished nonterminal)
  - R, a finite set of *rules* or *productions* of the form
    $$A \longrightarrow \beta$$
    $A \in N$ is a single nonterminal—hence, context free
    $\beta \in (\Sigma \cup N)^*$ is a finite string of terminals and nonterminals
  - Combine $A \longrightarrow \beta_i$ and $A \longrightarrow \beta_j$ into $A \longrightarrow \beta_i | \beta_j$
- Direct derivation, i.e., via a single application of a rule
  - From $(\Sigma \cup N)^*$ to $(\Sigma \cup N)^*$
  - $\delta_i \Rightarrow \delta_j$, meaning δ_i derives or yields δ_j
  - Given $A \longrightarrow \beta$, we get $\alpha A \gamma \Rightarrow \alpha \beta \gamma$
- Derivation over zero or more rule applications
  - $\Rightarrow^*$: reflexive, transitive closure of $\Rightarrow$
  - $\alpha_1 \Rightarrow^* \alpha_m$, through $m - 1$ direct derivations
  - Each derivation represents one snippet of possibilities

Context-Free Language

- Language generated from grammar $G = \langle N, \Sigma, R, S \rangle$

$$\mathscr{L}_G = \{w | w \in \Sigma^* \text{ and } S \Rightarrow^* w\}$$

  - Whatever can be derived from the start symbol
  - That ends up getting rid of all nonterminals
- Any such *generated* string of terminals, w above, is *grammatical* and is in the language
- Every other string of terminals is not grammatical and is not in the language
- A finite, ideally small, grammar should generate a large language
  - Capture the legitimate variations of use
  - Exclude the illegitimate variations
- Focuses on strings that are output
  - Doesn't reflect phrase structure in what is generated
  - Meaning is based on the invisible structure

CFG Example Sentence: I prefer a morning flight

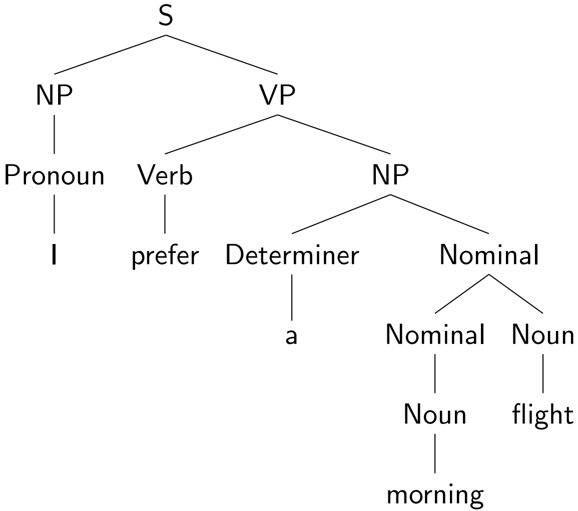
- Initial grammar and lexicon to derive the above sentence

  S $\longrightarrow$ NP VP
  NP $\longrightarrow$ Pronoun | Determiner Nominal
  VP $\longrightarrow$ Verb NP
  Nominal $\longrightarrow$ Nominal Noun | Noun

  Pronoun $\longrightarrow$ I
  Verb $\longrightarrow$ prefer
  Determiner $\longrightarrow$ a
  Noun $\longrightarrow$ morning | flight

- Why not have S $\longrightarrow$ N VP or S $\longrightarrow$ Pronoun VP?
- Need recursion, which the Nominal production gives us
- For additional sentences, we could insert

  VP $\longrightarrow$ VP NP PP (leaving Boston in the morning)
  VP $\longrightarrow$ VP PP (leaving in the morning)
  PP $\longrightarrow$ Preposition NP (from Boston)


S
NP
VP
Pronoun
Verb
NP
I
prefer
Determiner
Nominal
a
Nominal
Noun
Noun
flight
morning

Draw a Parse Tree

I prefer leaving Boston in the morning

Sentences in English

- Declarative $\sim$ default form
  - Subject NP ("I")
- Imperative, S $\longrightarrow$ VP
  - Usually, lack a subject "Go there"
  - But not always "You go there"
  - Subject *deletion* under a view that there is a subject
- Yes-no question, S $\longrightarrow$ Aux NP VP
  - Begin with auxiliary verb
  - Retain a main verb
- Wh-structures
  - In modern English, who, whose, when, where, what, which, how, why
  - Contain a wh-phrase

Wh Structures

- Wh-subject question, S $\longrightarrow$ Wh-NP VP
  - What airlines fly from Burbank to Denver?
  - The wh-phrase yields the subject
  - Wh-NP $\longrightarrow$ Wh-Pronoun (who, whom, whose, which)
  - Wh-NP $\longrightarrow$ Wh-Determiner NP (what, which)
- Wh-non-subject question, S $\longrightarrow$ Wh-NP Aux NP VP
  - What flights do you have from Burbank to Denver?
  - The wh-phrase is not the subject of the sentence, which is something else
  - Long-distance dependencies

Long-Distance Dependencies

- Consider the relationship indicated in our example and a possible (stylized) answer
  - What flights do you have from Burbank to Denver?
  - I have AA 999 from Burbank to Denver
  - There is an apparent discontinuity
- Semantic approach: Detect the relationship during interpretation

Long-Distance Dependencies

Syntactic approach: Understand the construction as phrase *movement*

- A *trace* or *empty category* is left behind ($\mathbb{t}$ below)
- Now a simple rule "want to ⇒ wanna" explains our earlier examples
  - Who does Vicky want to vote for $\mathbb{t}$?
    (Contraction applies)
    ⇒ Who does Vicky wanna vote for?
  - Who does Vicky want $\mathbb{t}$ to win?
    (Contraction doesn't apply: "want $\mathbb{t}$ to" doesn't match "want to")
    ⇒ *Who does Vicky wanna win

Evaluate a Grammar

Example sentence: I prefer a morning flight

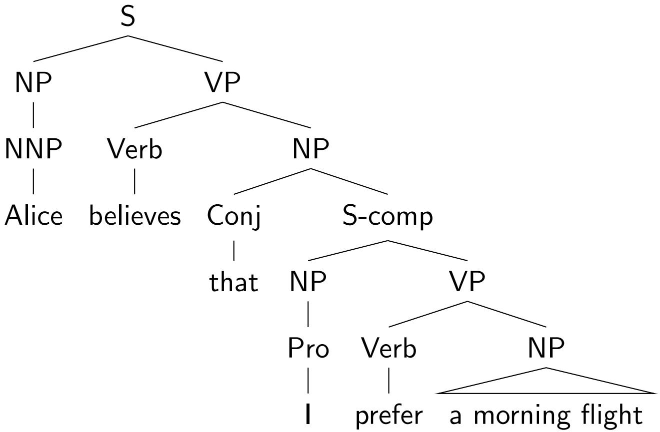
$$
\boxed{\begin{array}{l}
\mathrm{S} \longrightarrow \mathrm{X\ Y} \\
\mathrm{X} \longrightarrow \mathrm{Pronoun\ Verb\ Determiner} \\
\mathrm{Y} \longrightarrow \mathrm{NP} \mid \mathrm{NP\ NP} \\
\mathrm{NP} \longrightarrow \mathrm{Pronoun} \mid \mathrm{Nominal} \\
\mathrm{Nominal} \longrightarrow \ldots
\end{array}}
$$

- Assume the above grammar gives us the same coverage in terms of acceptable sentences and avoids all unacceptable sentences
- Is the grammar satisfactory? If so, how? If not, why not?

Clause: (Quasi) Sentence Expressing a Complete Thought

A node *S* in the parse tree that dominates all of the *arguments* of its main verb

- Alice believes that I prefer a morning flight
- Joe suggested that I prefer a morning flight



Finite and Nonfinite clauses

- Finite clauses have a verb that is tensed
  - Indicate a definite time when the event specified by the verb occurs
  - Indicate an instance of the event
- Nonfinite clauses may carry tense but not in the same way
  - Indicate a general occurrence of the specified event, not that it occurred specifically
  - Enable making generic *habitual* statements: Alice recommends stirring while you reheat the syrup
    - Gerunds, as in *-ing* verbs: stirring the pot
    - Infinitives, as in *to X*: to leave the lid off
    - Past participle, as in *-ed* verbs: to have preheated the oven Bob avoids to have begun before noon

Noun Phrases: Determiners and Predeterminers

- Determiners: not applied on mass nouns
  - Articles: A, an, the
  - Demonstratives: This, those, ...
  - Genitives: Det $\longrightarrow$ NP 's (notice recursion with NP)
    - Denver's mayor's mother's canceled flight
- Predeterminers: precede a determiner
  - All: All the king's men
  - A few of: A few of the king's men

Noun Phrases—Nominals: 1

- *Head noun*: The main component of an NP
- Before the head noun
  - Cardinals: Three friends; three and a half pounds; 3.14159 radians
  - Ordinals: The first one; the other flight
  - Quantifiers: Many students; some confused users
  - Adjective phrases (APs)
    - Quantifiers: Some confused users
    - With adverbs: The least expensive fare

Noun Phrases—Nominals: 2

- After the head noun: *postmodifiers*
  - Prepositional phrases: (all flights) from Cleveland
  - Nonfinite postmodifier clauses
    - Gerundive postmodifiers: Two flights arriving on Thursday
    - Infinitival postmodifiers: The last flight to arrive
    - Past participle postmodifiers: The aircraft used for this flight
  - (Restrictive) relative postmodifier clauses: A flight that serves breakfast
    - Relative pronouns (that, who): A flight that leaves on Sunday

Verb Phrases

A verb plus

- Nothing (*intransitive verb*): sleep
- NP: (prefer) a morning flight
- NP PP: (leave) Boston in the morning
- PP PP: (go) from Boston to Miami
- PP PP PP: (go) from Boston to Miami on a bus
- PP: (leaving) on Thursday
- Nonfinite VP: (want) to fly to San Francisco
- S (*Sentential complement*): (believes) AA 99 leaves from Boston

Major Verb Categories

Each verb can fit in only some of the VPs introduced above

- Traditionally
  - Intransitive
  - Transitive
  - Ditransitive
- The above don't tackle the subtle variations in language
- *Subcategorizing for* what kind of complement
- Yields a *subcategorization frame* or set of acceptable complements for each verb, e.g.,
  - NP
  - NP or nonfinite VP
  - Sentential complement
- *Complement*: phrase (word, clause) needed to complete an expression
  - Map to arguments in the obvious logical form understood from a phrase

Challenge in CFGs

- We can get hundreds (just for verbs) of lexical categories reflected as nonterminals with associated rules
  - VP $\longrightarrow$ Verb-with-NP-comp NP
  - VP $\longrightarrow$ Verb-with-S-comp S
  - Verb-with-NP-comp NP $\longrightarrow$ find | leave | repeat | ...
  - Verb-with-S-comp S $\longrightarrow$ think | believe | say | ...
- Enormous knowledge engineering (including maintenance) task
- Risks loss of generality
- Motivation for alternative representations to CFGs
  - Feature grammars: data driven by specifying lexical entries modularly

Coordination or Conjunction

And, or, but, ...

- *Coordinate*: composite phrase of two phrases separated by a conjunction
  - Also list enumerations
  - The conjoined phrases are of the same category
  - Evidence for the existence of a constituent structure
- NP and NP
  - the flights and the costs
- Nominal and Nominal
  - the flights and costs
- VP and VP
  - Departing Boston and arriving in Miami
- S and S
  - I like coffee and I like icecream
- AP and AP
  - Big and red

Treebanks

Especially, Penn Treebank

- Corpus of sentences
  - Parsed into trees
  - Represented in a standardized representation based on nested brackets or parentheses
  - Includes traces (shown as -*NONE*- with a numeric identifier)
- A treebank is an implicit grammar
  - Each upper node expands into its children
- Penn Treebank demonstrates a flat structure
  - Long rules, e.g., VP $\longrightarrow$ VBP PP PP PP PP PP ADVP PP
  - Many rules: 4,500 for VP and 17,500 in all for the Wall Street Journal corpus ($\sim$ 1M sentences)
  - May not be great for generalization

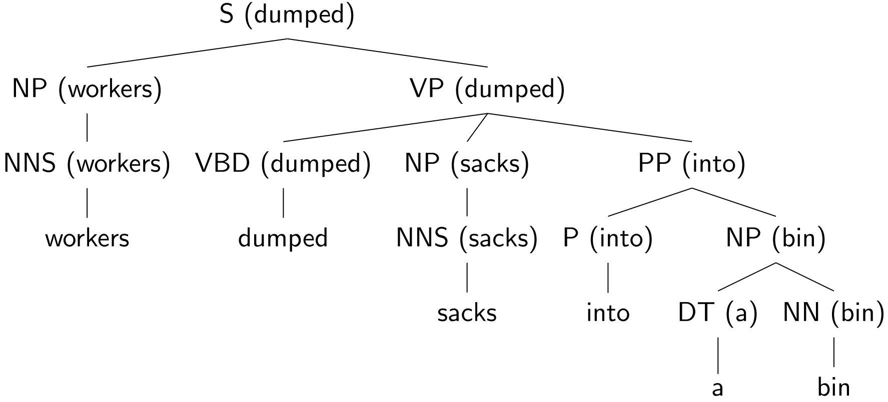
Heads

The grammatically central lexical part of a syntactic constituent

- Whatever predicate we have applies to the head
  - Olive oil is a kind of oil
  - A tall tree is a tree
  - To quickly swim is to swim
- Potentially augment a CFG
  - Identify headword for each production
  - Nontrivial and controversial, e.g., whether
    - To swim $\Rightarrow$ swim
    - To swim $\Rightarrow$ to
- Identify heads heuristically by first parsing and them walking a parse tree
  - The POS of the last word if it matches
  - Search for specific nodes right to left or left to right

Example Lexicalized (Head-Augmented) Tree

Collins' heuristic approach



Grammar Equivalence and Normal Form

- Weak equivalence: generate the same strings
- Strong equivalence
  - Weak plus assign the same phrase structure (up to renaming of nonterminals)
- Chomsky Normal Form, in which productions are of these forms:
  - Two at a time: A $\longrightarrow$ B C
  - Single terminal: A $\longrightarrow$ a
  - Not generating the empty string: Exclude A $\longrightarrow \varepsilon$
- Can convert from arbitrary CF grammar to Chomsky Normal Form that is weakly equivalent
  - Step used in the parsing algorithm

Converting to Chomsky Normal Form

- Conversion can increase or decrease the grammar size (number of productions)

$$
\begin{array}{rcl}
\text{VP} & \longrightarrow & \text{VP PP} \\
\text{VP} & \longrightarrow & \text{VBD NP PP} \\
\hline
\multicolumn{3}{l}{\text{is equivalent to}} \\
\hline
\text{VP} & \longrightarrow & \text{VBD X} \\
\text{VP} & \longrightarrow & \text{VP PP} \\
\text{X} & \longrightarrow & \text{NP PP} \\
\hline
\multicolumn{3}{l}{\text{is more general than}} \\
\hline
\text{VP} & \longrightarrow & \text{VBD NP PP} \\
\text{VP} & \longrightarrow & \text{VBD NP PP PP} \\
\text{VP} & \longrightarrow & \text{VBD NP PP PP PP} \\
\text{VP} & \longrightarrow & \text{VBD NP PP PP PP PP} \\
\ldots & &
\end{array}
$$

- Jurafsky claims equivalence but the smaller grammar is strictly more general because it finitely expresses unbounded repetitions of PP

Examples of Chomsky Normal Form

State a grammar and an equivalent CNF grammar that is strictly smaller (has fewer productions)

Lexicalized Grammars

Categorial grammar being one such

- Address the redundancy and brittleness of CFGs
- Greater emphasis on lexical knowledge
- Data driven in having smaller grammars that go over more extensive lexicons
- Improve modularity
- Can handle changing word usage and new words

Categorial Grammar

Motivated by composition in the spirit of the lambda calculus
Components: categories, lexicon, combination rules

- Set of categories
  - Atomic categories: noun, sentence, ...
  - X/Y: function from category Y (on the right) to category X
  - X\Y: function from category Y (on the left) to category X
- Lexicon that associates words with categories, atomic or functional
  - John: NNP (singular proper noun)
  - Water: NN (singular or mass noun)
  - Drinks, as a transitive verb: (S\NNP)/NN
- Set of rules governing how categories combine (in context)
  - Forward function application: X/Y Y $\Rightarrow$ X
  - backward function application: X\Y Y $\Rightarrow$ X
  - X CONJ X $\Rightarrow$ X: Y X\Y $\Rightarrow$ X

Example Derivation Tree

$$\cfrac{\cfrac{\text{John}}{\text{NNP}} \qquad \cfrac{\cfrac{\text{drinks}}{(\text{S}\backslash\text{NNP})/\text{NN}} \qquad \cfrac{\text{water}}{\text{NN}}}{S\backslash NNP}{>}}{S}{<}$$

- Shown top to bottom
- Line demarcates scope of the category listed below it
- $>$ and $<$ indicate which is the function and which is the argument

Example Derivation Tree with Conjunction

$$
\begin{array}{ccccc}
\text{John} & \text{drinks} & \text{or} & \text{wastes} & \text{water} \\
\hline
\mathrm{NNP} & \mathrm{(S\backslash NNP)/NN} & \mathrm{conj} & \mathrm{(S\backslash NNP)/NN} & \mathrm{NN} \\
\cline{2-4}
 & \multicolumn{3}{c}{(S\backslash NNP)/NN \quad {<}\Phi{>}} & \\
\cline{2-5}
 & \multicolumn{4}{c}{S\backslash NNP \quad >} \\
\hline
\multicolumn{5}{c}{S \quad <}
\end{array}
$$

CCG: Combinatory Categorial Grammar

Using the same lexical entries to produce new combinations

- Forward composition (signified by $>\mathbf{B}$): $X/Y \;\; Y/Z \Rightarrow X/Z$
- Backward composition (signified by $<\mathbf{B}$): $Y\backslash Z \;\; X\backslash Y \Rightarrow X\backslash Z$
  - "Cancel" out the middle Y in both forward and backward composition
- Type raising (arguments to the right, signified by $>\mathbf{T}$): $X \Rightarrow T/(T\backslash X)$
- Type raising (arguments to the left, signified by $<\mathbf{T}$): $X \Rightarrow T\backslash(T/X)$
  - Example: $NP \Rightarrow S/(S\backslash NP)$

Original Derivation

$$
\begin{array}{ccc}
\text{John} & \text{drinks} & \text{water} \\
\hline
\mathrm{NNP} & \mathrm{(S\backslash NNP)/NN} & \mathrm{NN} \\
 & \multicolumn{2}{c}{\overline{\qquad\qquad\qquad\qquad}{>}} \\
 & \multicolumn{2}{c}{S\backslash NNP} \\
\multicolumn{3}{c}{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}{<}} \\
\multicolumn{3}{c}{S}
\end{array}
$$

Type Raising Derivation

$$
\begin{array}{ccc}
\text{John} & \text{drinks} & \text{water} \\
\hline
\mathrm{NNP} & \mathrm{(S\backslash NNP)/NN} & \mathrm{NN} \\
\overline{\qquad\qquad}{>}\mathbf{T} & & \\
S/(S\backslash NNP) & & \\
\multicolumn{2}{c}{\overline{\qquad\qquad\qquad\qquad}{>}\mathbf{B}} & \\
\multicolumn{2}{c}{S/NNP} & \\
\multicolumn{3}{c}{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}{>}} \\
\multicolumn{3}{c}{S}
\end{array}
$$

Benefits of CCG

- Supports incremental interpretation (left to right in English), which may have some psychological realism
- Supports coordinating (conjoining) phrases that aren't obvious constituents
  Billy eats <u>icecream for dinner</u> and <u>salad for dessert</u>
- For brevity, write VP for S\NP
- Type of "eats": (VP/PP)/NP
- Raise type of "icecream" ($\sim$ Y\Z): NP $\Rightarrow$ (VP/PP)\((VP/PP)/NP)
- Raise type of "for dinner" ($\sim$ X\Y): PP $\Rightarrow$ VP\(VP/PP)
- Backward compose the raised types (Y\Z X\Y $\Rightarrow$ X\Z)
  Y binds to (VP/PP) and is discarded, yielding VP\((VP/PP)/NP)
- Likewise, "salad for dessert" yields VP\((VP/PP)/NP)
- Conjoin these to obtain VP\((VP/PP)/NP)
- Apply on "eats" to obtain VP ($\equiv$ S\NP)
- Apply on "Billy" (NP) to obtain S

Long-Distance Dependencies in CCG

- A transitive verb ("ate") expects
  - Subject NP ("Billy") to its left
  - Object NP ("the salad") to its right
- Here, the object NP is moved to the front
- Notice that "Billy ate" is of type S/NP
- The main work is done by "that" by mapping
  - S/NP (needs an NP to its right) to
  - NP\NP (takes an NP to its left and yields an NP)

$$\begin{array}{ccccc}
\text{The} & \text{salad} & \text{that} & \text{Billy} & \text{ate}\\
\hline
\mathrm{NP/N} & \mathrm{N} & \mathrm{(NP\backslash NP)/(S/NP)} & \mathrm{NP} & \mathrm{(S\backslash NP)/NP}\\
\cline{1-2}\cline{4-4}
\multicolumn{2}{c}{NP \quad >} & & S\backslash(S/NP) \quad >\mathbf{T} & \\
\cline{4-5}
 & & & \multicolumn{2}{c}{S/NP \quad >\mathbf{B}}\\
\cline{3-5}
 & & \multicolumn{3}{c}{NP\backslash NP \quad >}\\
\hline
\multicolumn{5}{c}{NP \quad <}
\end{array}$$