Faculty for Informatics

Technical University of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

Social Computing
Research Group

Chapter 12
Constituency Grammars

- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

Constituents

Constituents - groups of words behaving as a single units

- example: noun phrases:

Harry the Horse	a high-class spot such as Mindy's
the Broadway coppers	the reason he comes into the Hot Box
they	three parties from Brooklyn

- evidence that these are constituents: can appear before verbs

three parties from Brooklyn *arrive...*
a high-class spot such as Mindy's *attracts...*
the Broadway coppers *love...*
they *sit*

while parts thereof cannot:

*from *arrive...*	*as *attracts...*
*the *is...*	*spot *sat...*

Constituents

- other evidence that indicate constituency: different positions possible as a whole: example: prepositional phrase:

On September seventeenth, I'd like to fly from Atlanta to Denver
I'd like to fly *on September seventeenth* from Atlanta to Denver
I'd like to fly from Atlanta to Denver *on September seventeenth*

while parts thereof cannot appear in different positions individually :

*On September, I'd like to fly seventeenth from Atlanta to Denver
*On I'd like to fly September seventeenth from Atlanta to Denver
*I'd like to fly on September from Atlanta to Denver seventeenth

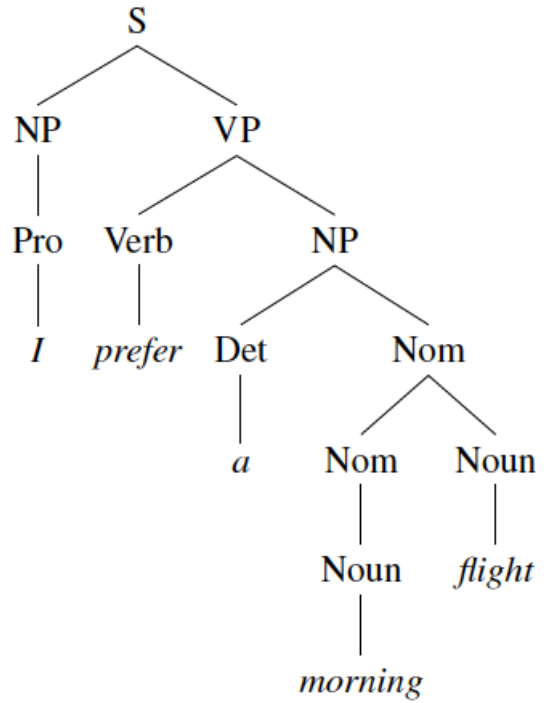
- Formal Grammars provide rules on how to construct and use constituents

Context Free Grammars - Example

Noun	→	*flights* | *breeze* | *trip* | *morning* | *flight*
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	→	*cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun	→	*me* | *I* | *you* | *it*
Proper-Noun	→	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner	→	*the* | *a* | *an* | *this* | *these* | *that*
Preposition	→	*from* | *to* | *on* | *near*
Conjunction	→	*and* | *or* | *but*

The lexicon for $\mathcal{L}_0$.

simple language $\mathcal{L}_0$
for flight booking



[$_S$ [$_{NP}$ [$_{Pro}$ I]] [$_{VP}$ [$_V$ prefer] [$_{NP}$ [$_{Det}$ a] [$_{Nom}$ [$_N$ morning] [$_{Nom}$ [$_N$ flight]]]]]]

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

The grammar for $\mathcal{L}_0$, with example phrases for each rule.

CF Grammars – Formal Definitions

- Context Free (Phrase Structure) Grammar is 4-tuple of

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S	a designated **start symbol** and a member of N

here: terminal symbols are words

Notation:

Capital letters like A, B, and S	Non-terminals
S	The start symbol
Lower-case Greek letters like α, β, and γ	Strings drawn from $(\Sigma \cup N)*$
Lower-case Roman letters like u, v, and w	Strings of terminals

CF Grammars – Formal Definitions

- if $A \rightarrow \beta$ is a production of R and α and γ are any strings in the set $(\Sigma \cup N)*$, then we say that $\alpha A \gamma$ **directly derives** $\alpha \beta \gamma$, or $\alpha A \gamma \Rightarrow \alpha \beta \gamma$.

- Let $\alpha_1, \alpha_2, \ldots, \alpha_m$ be strings in $(\Sigma \cup N)*, m \geq 1$, such that

$$\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \ldots, \alpha_{m-1} \Rightarrow \alpha_m$$

  We say that α_1 **derives** α_m, or $\alpha_1 \stackrel{*}{\Rightarrow} \alpha_m$.

- the language that a CF grammar generates:

$$\mathcal{L}_G = \{w | w \text{ is in } \Sigma* \text{ and } S \stackrel{*}{\Rightarrow} w\}$$

- Syntactic Parsing: given sequence ("string") of words & CF grammar: construct parse tree

Examples

- declarative sentences:

$S \rightarrow NP\ VP$ I prefer a morning flight

- imperative sentences:

$S \rightarrow VP$ Show the lowest fare

- yes-no-questions:

$S \rightarrow Aux\ NP\ VP$ Do any of these flights have stops?

Examples

- Wh-subject-questions:

  $S \rightarrow Wh\text{-}NP\ VP$ What airlines fly from Burbank to Denver?

- Wh-non-subject-questions:

  $S \rightarrow Wh\text{-}NP\ Aux\ NP\ VP$

  What flights do you have from Burbank to Tacoma Washington?

  - Wh-NP *What flights* is not subject of sentence
  - Wh-NP has a long-range dependency to VP *have* that it relates to:
    - represent as semantic relation
    - represent as syntactic relation

The Noun Phrase

Noun phrase: Proper Nouns, pronouns, $NP \rightarrow Det \ \ Nominal$ constructions

- Determiners:

lexical determiners:

a stop	the flights	this flight
those flights	any flights	some flights

$Det \rightarrow NP \ 's$:

United's flight
United's pilot's union
Denver's mayor's mother's canceled flight

mass nouns / nouns in plural: should be / can be without determiner:

Does this flight serve dinner?
Show me *flights* from San Francisco to Denver on weekdays

The Noun Phrase

- Nominal:
  either simple (head-)noun:

  $Nominal \rightarrow Noun$

- or with pre-head noun modifiers:
  cardinal numbers, ordinal numbers, quantifiers, and adjectives or adjective phrases

two friends one stop

the first one the next day the second leg
the last flight the other American flight

many fares

a *first-class* fare a *non-stop* flight
the *longest* layover the *earliest* lunch flight

the *least expensive* fare

The Noun Phrase

- or with post-head noun modifiers (postmodifiers); three common kinds:

prepositional phrases	all flights *from Cleveland*
non-finite clauses	any flights *arriving after eleven a.m.*
relative clauses	a flight *that serves breakfast*

  - prepositional phrase (PP) postmodifiers

$$Nominal \rightarrow Nominal\ PP$$

all flights *[from Cleveland] [to Newark]*
arrival *[in San Jose] [before seven p.m.]*
a reservation *[on flight six oh six] [from Tampa] [to Montreal]*

  - non-finite postmodifiers
    - gerundive (-ing) form
    - -ed form
    - infinitive form

The Noun Phrase

- gerundive (-ing) form postmodifiers

any of those *[leaving on Thursday]*
any flights *[arriving after eleven a.m.]*
flights *[arriving within thirty minutes of each other]*

$$\begin{aligned} \textit{Nominal} &\rightarrow \textit{Nominal GerundVP} \\ \textit{GerundVP} &\rightarrow \textit{GerundV NP} \\ &\quad | \quad \textit{GerundV PP} \mid \textit{GerundV} \mid \textit{GerundV NP PP} \\ \textit{GerundV} &\rightarrow \textit{being} \mid \textit{arriving} \mid \textit{leaving} \mid \ldots \end{aligned}$$

- -ed form or infinitive form postmodifiers:

the last flight *to arrive in Boston*
I need to have dinner *served*
Which is the aircraft *used by this flight*?

The Noun Phrase

- postnominal relative clause (restrictive relative clause):

a flight *that serves breakfast*
flights *that leave in the morning*
the one *that leaves at ten thirty five*

$$Nominal \rightarrow Nominal\ RelClause$$
$$RelClause \rightarrow (who \mid that)\ VP$$

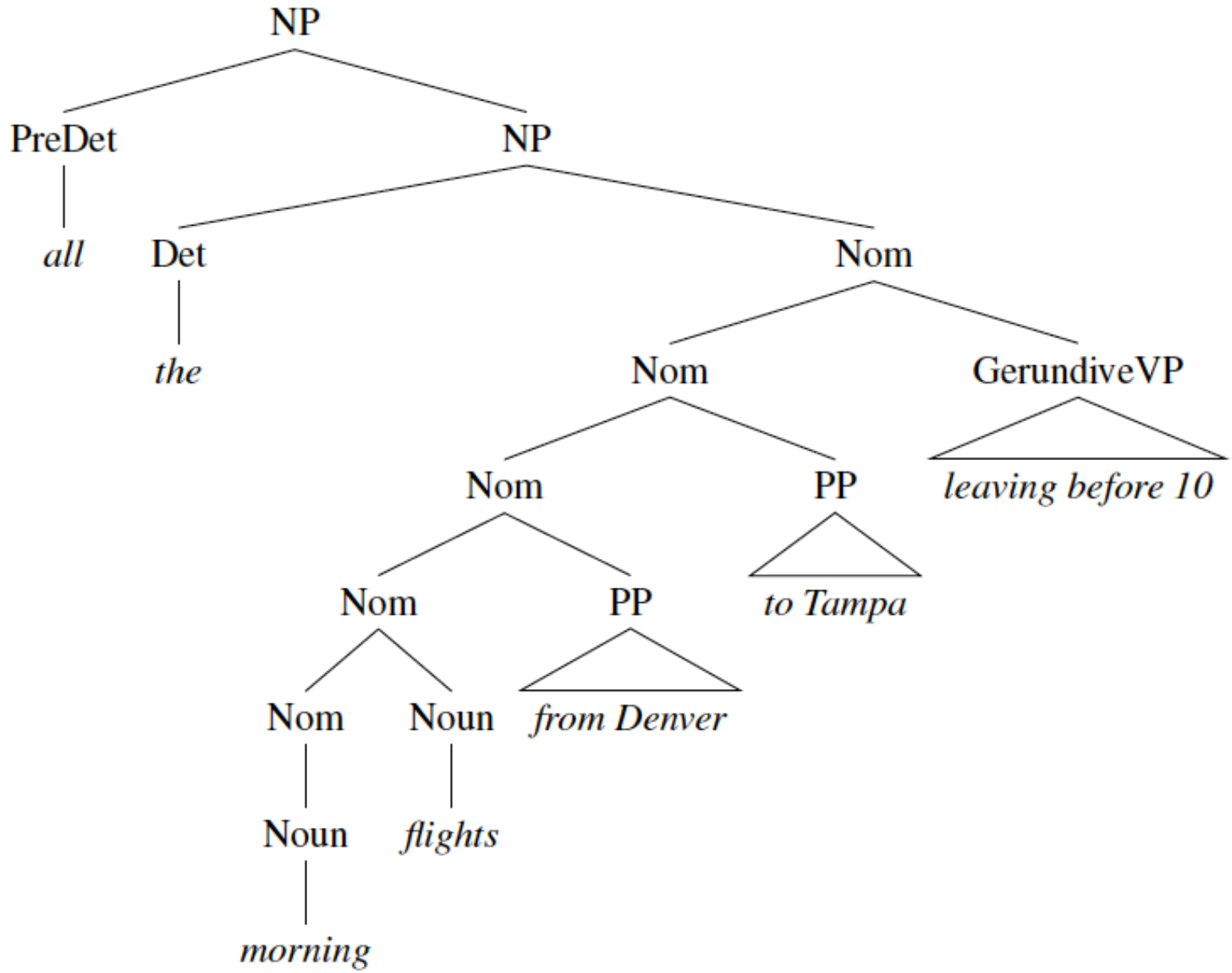
postnominal modifiers can also be combined:

a flight *[from Phoenix to Detroit] [leaving Monday evening]*
evening flights *[from Nashville to Houston] [that serve dinner]*
a friend *[living in Denver] [that would like to visit me here in Washington DC]*

also possible Predeterminers before the whole noun phrase:

all the flights all flights all non-stop flights

The Noun Phrase – Larger Example



The Verb Phrase

- example: simple rules

  $VP \rightarrow Verb$ disappear
  $VP \rightarrow Verb\ NP$ prefer a morning flight
  $VP \rightarrow Verb\ NP\ PP$ leave Boston in the morning
  $VP \rightarrow Verb\ PP$ leaving on Thursday

- example: also possible: sentences as complements for verbs (sententinal complements)

  $VP \rightarrow Verb\ S$

  You [$_{VP}$ [$_{V}$ said [$_{S}$ you had a two hundred sixty six dollar fare]]
  [$_{VP}$ [$_{V}$ Tell] [$_{NP}$ me] [$_{S}$ how to get from the airport in Philadelphia to down-town]]
  I [$_{VP}$ [$_{V}$ think [$_{S}$ I would like to take the nine thirty flight]]

- example: also possible: VPs as complements for verbs

  I want [$_{VP}$ to fly from Milwaukee to Orlando]
  Hi, I want [$_{VP}$ to arrange three flights]

The Verb Phrase

- idea subcategorize verbs:
  - old: transitive verbs, intransitive verbs
  - modern: over 100 subcategorization frames (complements):

Frame	Verb	Example
$\emptyset$	eat, sleep	I ate
NP	prefer, find, leave	Find [$_{NP}$ the flight from Pittsburgh to Boston]
NP NP	show, give	Show [$_{NP}$ me] [$_{NP}$ airlines with flights from Pittsburgh]
PP_{from} PP_{to}	fly, travel	I would like to fly [$_{PP}$ from Boston] [$_{PP}$ to Philadelphia]
NP PP_{with}	help, load	Can you help [$_{NP}$ me] [$_{PP}$ with a flight]
VPto	prefer, want, need	I would prefer [$_{VPto}$ to go by United airlines]
VPbrst	can, would, might	I can [$_{VPbrst}$ go from Boston]
S	mean	Does this mean [$_{S}$ AA has a hub in Boston]

not every verb supports all of these → possible:

Verb-with-NP-complement → *find* | *leave* | *repeat* | ...

Verb-with-S-complement → *think* | *believe* | *say* | ...

Verb-with-Inf-VP-complement → *want* | *try* | *need* | ...

VP → *Verb-with-no-complement* disappear

VP → *Verb-with-NP-comp NP* prefer a morning flight

VP → *Verb-with-S-comp S* said there were two flights

Coordinations

- Conjoin major phrase types (noun phrases, verb phrases etc.) with conjunctions (*and, or, but, …*) → coordinate phrases.
  Examples with and:

$NP \rightarrow NP\ and\ NP$

Please repeat [$_{NP}$ [$_{NP}$ the flights] *and* [$_{NP}$ the costs]]
I need to know [$_{NP}$ [$_{NP}$ the aircraft] *and* [$_{NP}$ the flight number]]

$Nominal \rightarrow Nominal\ and\ Nominal$

Please repeat the [$_{Nom}$ [$_{Nom}$ flights] *and* [$_{Nom}$ costs]]
I need to know the [$_{Nom}$ [$_{Nom}$ aircraft] *and* [$_{Nom}$ flight number]]

$VP \rightarrow VP\ and\ VP$

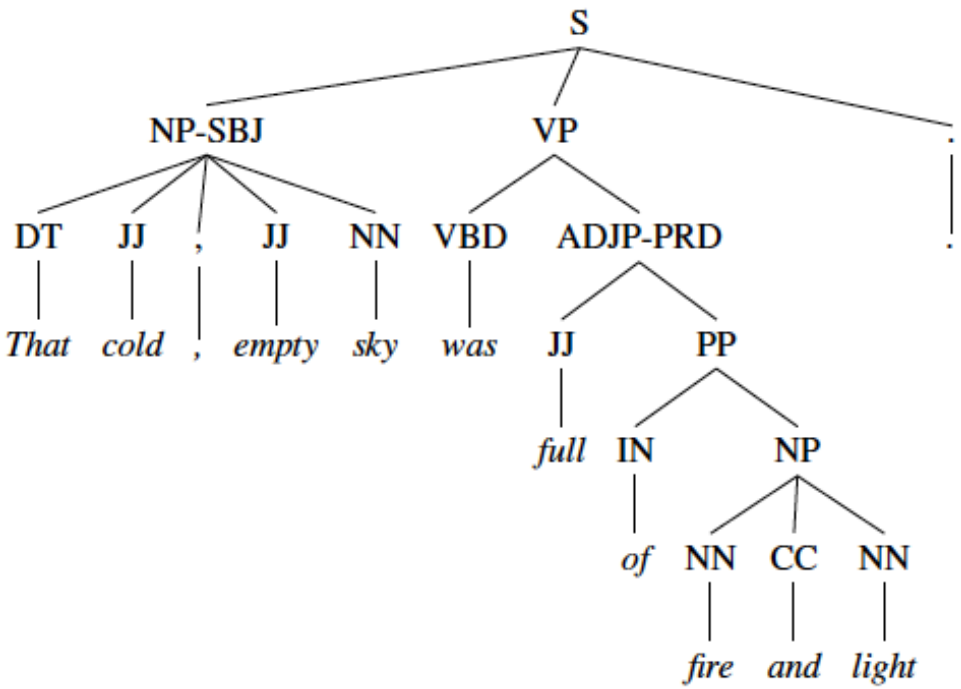
$S \rightarrow S\ and\ S$

What flights do you have [$_{VP}$ [$_{VP}$ leaving Denver] *and* [$_{VP}$ arriving in San Francisco]]
[$_{S}$ [$_{S}$ I'm interested in a flight from Dallas to Washington] *and* [$_{S}$ I'm also interested in going to Baltimore]]

Treebanks

- Corpora with parse trees for each sentence. Example: Penn Treebank

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```



- different treebank or different parts of a treebank may use slightly different notation or non-terminals

Treebanks as Grammars

- From the following two sentences, the following rules may be derived

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those)(NNS assets)))))))))))))
    (, ,) ('' '')
    (NP-SBJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```

Grammar	Lexicon
$S \rightarrow NP\ VP\ .$	$PRP \rightarrow we \mid he$
$S \rightarrow NP\ VP$	$DT \rightarrow the \mid that \mid those$
$S \rightarrow$ " S ", $NP\ VP\ .$	$JJ \rightarrow cold \mid empty \mid full$
$S \rightarrow$ -NONE-	$NN \rightarrow sky \mid fire \mid light \mid flight \mid tomorrow$
$NP \rightarrow DT\ NN$	$NNS \rightarrow assets$
$NP \rightarrow DT\ NNS$	$CC \rightarrow and$
$NP \rightarrow NN\ CC\ NN$	$IN \rightarrow of \mid at \mid until \mid on$
$NP \rightarrow CD\ RB$	$CD \rightarrow eleven$
$NP \rightarrow DT\ JJ\ ,\ JJ\ NN$	$RB \rightarrow a.m.$
$NP \rightarrow PRP$	$VB \rightarrow arrive \mid have \mid wait$
$NP \rightarrow$ -NONE-	$VBD \rightarrow was \mid said$
$VP \rightarrow MD\ VP$	$VBP \rightarrow have$
$VP \rightarrow VBD\ ADJP$	$VBN \rightarrow collected$
$VP \rightarrow VBD\ S$	$MD \rightarrow should \mid would$
$VP \rightarrow VBN\ PP$	$TO \rightarrow to$
$VP \rightarrow VB\ S$	
$VP \rightarrow VB\ SBAR$	
$VP \rightarrow VBP\ VP$	
$VP \rightarrow VBN\ PP$	
$VP \rightarrow TO\ VP$	
$SBAR \rightarrow IN\ S$	
$ADJP \rightarrow JJ\ PP$	
$PP \rightarrow IN\ NP$	

Treebanks as Grammars

- whole treebank: 17500 rule types

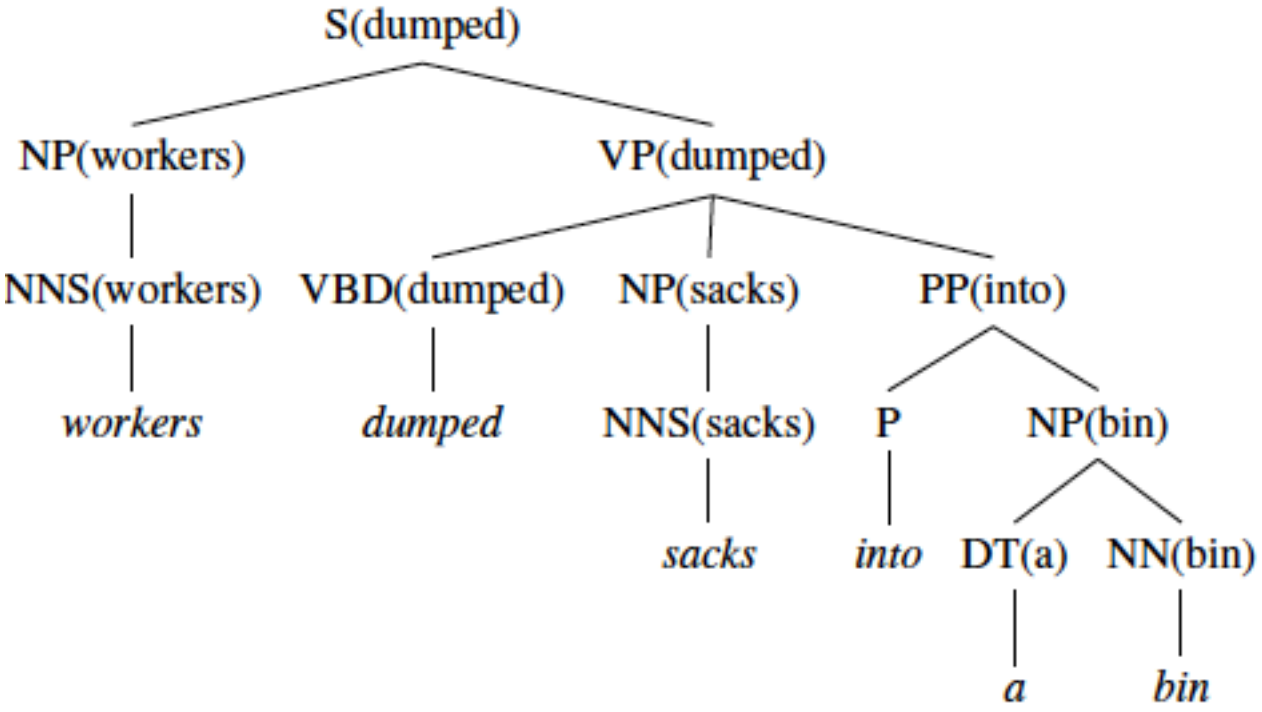
```
VP → VBD PP
VP → VBD PP PP
VP → VBD PP PP PP
VP → VBD PP PP PP PP
VP → VB ADVP PP
VP → VB PP ADVP
VP → ADVP VB PP
...
```

```
NP → DT JJ NN
NP → DT JJ NNS
NP → DT JJ NN NN
NP → DT JJ JJ NN
NP → DT JJ CD NNS
NP → RB DT JJ NN NN
NP → RB DT JJ JJ NNS
NP → DT JJ JJ NNP NNS
NP → DT NNP NNP NNP NNP JJ NN
NP → DT JJ NNP CC JJ JJ NN NNS
NP → RB DT JJS NN NN SBAR
NP → DT VBG JJ NNP NNP CC NNP
NP → DT JJ NNS , NNS CC NN NNS NN
NP → DT JJ JJ VBG NN NNP NNP FW NNP
NP → NP JJ , JJ `` SBAR '' NNS
...
```

Heads

- each constituent: associate with a lexical head (word):
  For a NP it is a N, for a VP it is a V

- augment rules and thus parse trees with head words:



Heads

- Finding heads, if heads are not explicitly specified in rules: non-trivial: parse sentence and apply head rules

- example: find head of an NP:

  - If the last word is tagged POS, return last-word.
  - Else search from right to left for the first child which is an NN, NNP, NNPS, NX, POS, or JJR.
  - Else search from left to right for the first child which is an NP.
  - Else search from right to left for the first child which is a $, ADJP, or PRN.
  - Else search from right to left for the first child which is a CD.
  - Else search from right to left for the first child which is a JJ, JJS, RB or QP.
  - Else return the last word

Grammar Equivalence and Normal Form

- two grammars are weakly equivalent if they generate the same language
- two grammars are strongly equivalent if they generate the same language and the same phrase structure (sequence of non-terminals up to renaming)

- a CF grammar is in Chomsky Normal Form (CNF) if it is ϵ-free and every rule is of the form A → BC or A → a.

$$A \rightarrow B\ C\ D \Longrightarrow \begin{array}{l} A \rightarrow B\ X \\ X \rightarrow C\ D \end{array}$$

$$\begin{array}{l} \text{VP} \rightarrow \text{VBD NP PP} \\ \text{VP} \rightarrow \text{VBD NP PP PP} \\ \text{VP} \rightarrow \text{VBD NP PP PP PP} \\ \text{VP} \rightarrow \text{VBD NP PP PP PP PP} \\ \ldots \end{array} \Longrightarrow \begin{array}{l} \text{VP} \rightarrow \text{VBD NP PP} \\ \text{VP} \rightarrow \text{VP PP} \end{array}$$

Lexicalized Grammars: Combinatory Categorial Grammars

- CF grammars may have too little an emphasis on the lexicon and may have too many actual non-terminal productions (e.g. compare subcategorization frames)

- → place emphasis on lexicon. Various approaches exist.

- example: Combinatory Categorial Grammar (CCG)

Combinatory Categorial Grammars

- define $\mathscr{C}$, a set of categories for a grammar
  - $\mathscr{A} \subseteq \mathscr{C}$, where $\mathscr{A}$ is a given set of atomic elements (e.g. NP)
  - $(X/Y), (X\backslash Y) \in \mathscr{C}$, if $X, Y \in \mathscr{C}$

- Rules:

$X/Y \;\; Y \;\Rightarrow\; X$ $\quad$ X/Y : function that seeks its argument to the right

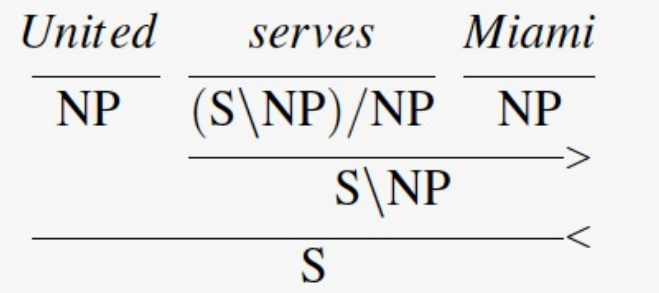
$Y \;\; X\backslash Y \;\Rightarrow\; X$ $\quad$ $X\backslash Y$: function that seeks its argument to the left

- Lexicon: assignment of words to categories

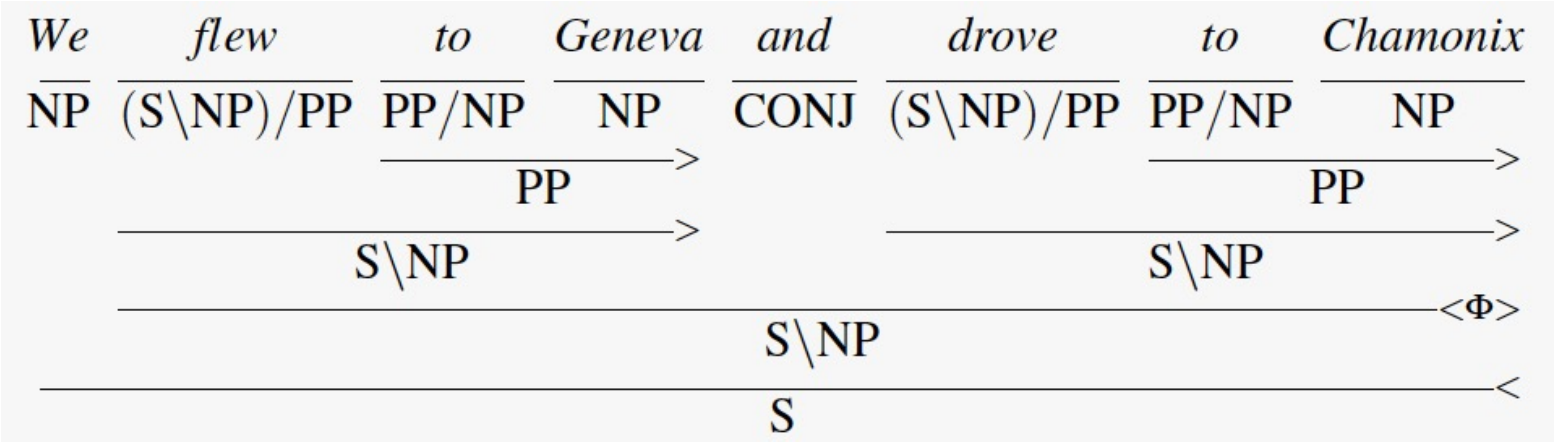
flight :	N	
Miami :	NP	
cancel :	$(S\backslash NP)/NP$	transitive verb (one object)
give :	$((S\backslash NP)/NP)/NP$	bitransitive verb (two objects)

Combinatory Categorial Grammars

- example:



- coordination (meta-)rule : $X \; CONJ \; X \Rightarrow X$



Combinatory Categorial Grammars

- more expressive power than CF by:
  - function composition:

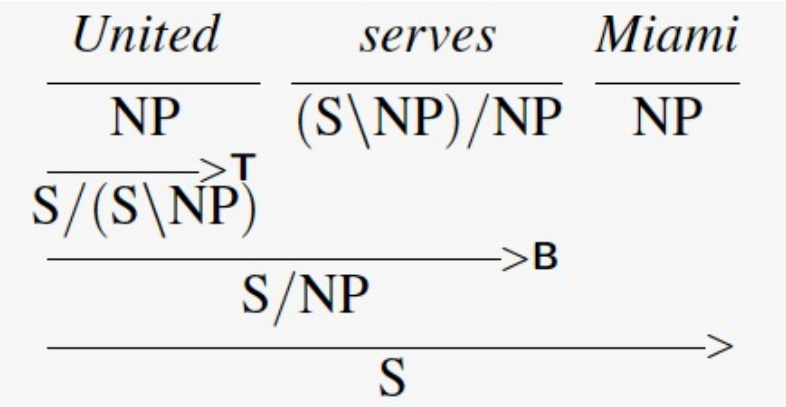
$$X/Y \;\; Y/Z \;\Rightarrow\; X/Z$$
$$Y\backslash Z \;\; X\backslash Y \;\Rightarrow\; X\backslash Z$$

  - type raising:

$$X \;\Rightarrow\; T/(T\backslash X)$$
$$X \;\Rightarrow\; T\backslash(T/X)$$

with T any category

- example for type raising (**T**) and function composition (**B**):
  type-raise NP: $NP \;\Rightarrow\; S/(S\backslash NP)$

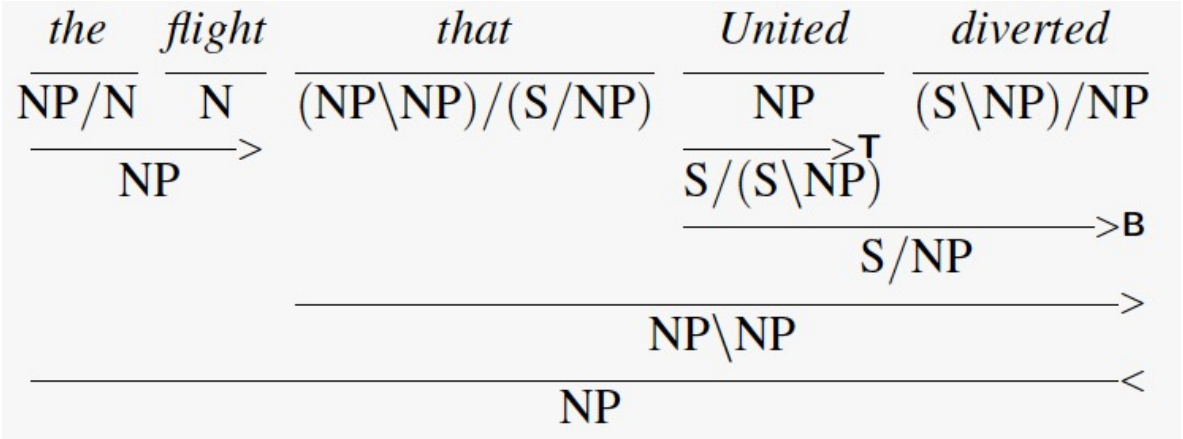


→ left to right parsing (human style) is possible

→ *United serves* as S/NP (not normally considered a constituent)

Combinatory Categorial Grammars

- modelling of long range dependencies is possible



(NP\NP) / (S/NP):
seeks a sentence missing an (NP)-argument to its right,
and transforms it into an NP seeking a missing (NP)-element to its left

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Oct 2019); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2019); this slide-set mainly related to chapter 12

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  == standard approach