CMPE 297, INSTRUCTOR: JORJETA JETCHEVA

GRAMMARS AND PARSING

GRAMMARS

- A grammar is a formal model of the syntax of natural language (or computer language)
  - Syntax is the arrangement of words to create well-formed sentences
- The study of grammar is not new – the grammar of Sanskrit was described by the Indian grammarian Panini between 4th and 7th century BCE

CONSTITUENCY OVERVIEW

- Syntactic constituency describes the idea that groups of words can behave as single units aka constituents
- A noun phrase (NP) is an example constituent
  - An NP is a sequence of words surrounding at least one noun

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

How do we know that a particular set of words forms a constituent?

- Constituents can appear in similar syntactic environments, which is not true for individual words within the constituent
  - e.g., a NP can appear before a verb, but its components words cannot
- Constituents can be part of preposed and postposed constructions
  - Propositional phrases can be placed at different locations within a sentence, e.g., the beginning (preposed) or end (postposed) but their individual words cannot be

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

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

asterisk (*) mark fragments that are not grammatical English sentences

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*

*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

CONTEXT-FREE GRAMMARS

CONTEXT-FREE GRAMMARS: OVERVIEW

- The most commonly used formal grammar system for constituent modeling is the Context-Free Grammar (CFG)
  - CFGs are also known as Phrase-Structure Grammars and Backus-Naur form (BNF)
  - Chomsky (1956) formalized the idea of a grammar focusing on constituents (Backus independently did so in 1959)
- A CFG has a
  - A lexicon of terminal symbols (words) and non-terminal symbols (lexical categories of parts of speech, e.g., NP)
  - Set of rules (aka productions) that describe how symbols of a language can be ordered and grouped
    - Each rule has a non-terminal symbol on the left-hand side, and an ordered list of terminal and non-terminal symbols on the right-hand side of the arrow
    - Below, | indicates "or", and → means "goes to"

$$
\begin{aligned}
NP &\rightarrow Det\ Nominal \\
NP &\rightarrow ProperNoun \\
Nominal &\rightarrow Noun \mid Nominal\ Noun
\end{aligned}
$$

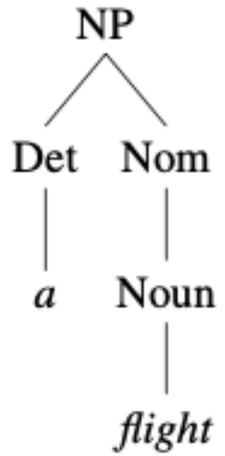
$$
\begin{aligned}
Det &\rightarrow a \\
Det &\rightarrow the \\
Noun &\rightarrow flight
\end{aligned}
$$

GENERATION OF SENTENCES

- A CFG can be used to generate sentences, and also to assign structure to a given sentence
- For example, using the rules below, we can generate the sentence "a flight" as follows:

$$
\begin{aligned}
NP &\rightarrow Det\ Nominal \\
NP &\rightarrow ProperNoun \\
Nominal &\rightarrow Noun \mid Nominal\ Noun \\
\\
Det &\rightarrow a \\
Det &\rightarrow the \\
Noun &\rightarrow flight
\end{aligned}
$$

So starting from the symbol:	*NP*
we can use our first rule to rewrite *NP* as:	*Det Nominal*
and then rewrite *Nominal* as:	*Det Noun*
and finally rewrite these parts-of-speech as:	*a flight*



- Derivations such as the above where the grammar is used to generate sequences of words (or strings) is commonly represented as a parse tree
- Each grammar has a designated start symbol (usually denoted by S which stands for sentence)
- The set of strings that can be derived from the start symbol are the formal language defined by the grammar
  - The derivable sentences are called "grammatical sentences", whereas those not derivable by the grammar are called "ungrammatical sentences".

EXAMPLE: AIR TRAFFIC INFORMATION SYSTEM (ATIS) LANGUAGE

- ATIS was a spoken language system for booking flights (1990)
- ATIS is captured by the $\mathscr{L}_0$ CFG (in the figures)
- A verb phrase (VP) is typically a verb followed by other types of words (e.g., "prefer a morning flight")
- A prepositional phrase (PP) is generally a preposition followed by a NP (e.g., "from Los Angeles")
- Derivations can be described with parse trees or bracketed representations of the parse tree

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

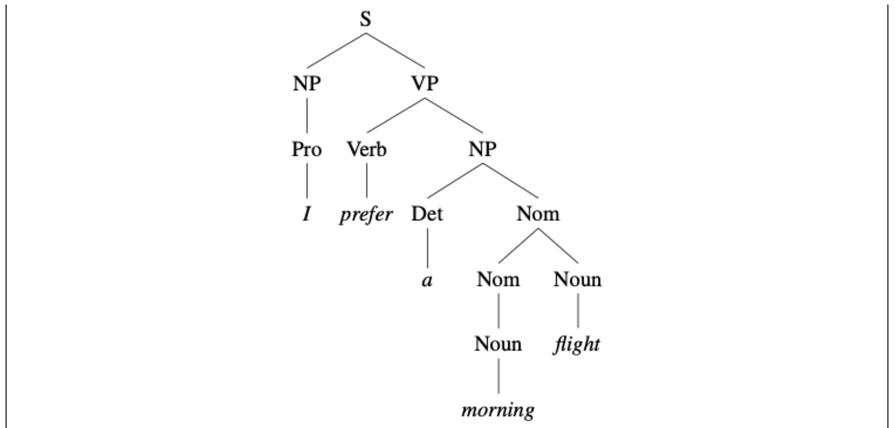


Figure 12.4 The parse tree for "I prefer a morning flight" according to grammar $\mathcal{L}_0$.

Noun	→	*flights* | *flight* | *breeze* | *trip* | *morning*
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly* | *do*
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* | *in*
Conjunction	→	*and* | *or* | *but*

Figure 12.2 The lexicon for $\mathcal{L}_0$.

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

Figure 12.3 The grammar for $\mathcal{L}_0$, with example phrases for each rule.

FORMAL DEFINITION OF CFG

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

- A context-free grammar G is defined by the 4-tuple: N, Σ, R, S
- One string derives another one if it can be rewritten as the second one by some series of rule applications (or productions)

Direct derivation:

if $A \rightarrow \beta$ is a production of R, and α and γ are strings in the set $(\Sigma \cup N)^*$, then we say that $\alpha A \gamma$ directly derives $\alpha\beta\gamma$ which is expressed as follows:

$\alpha A \gamma \implies \alpha\beta\gamma$ (we replace a non-terminal with a terminal)

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

Generalizing direct derivation:

if we have $\alpha_1, \alpha_2, \ldots, \alpha_m$ that are strings in $(\Sigma \cup N)^*$, and $m \geq 1$,

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

we say that α_1 derives α_m, or $\alpha_1 \overset{*}{\implies} \alpha_m$

- The language $\mathcal{L}_G$ generated by G is the set of strings composed of terminal symbols that can be derived from the designated start symbol can be expressed as follows: $\mathcal{L}_G = \{w \mid w \text{ is in } \Sigma^* \text{ and } S \overset{*}{\implies} w\}$

GRAMMAR EQUIVALENCE

Grammar equivalence

- Two grammars are **strongly equivalent** if they generate the same set of strings AND they assign the same phrase structure to each sentence
- Two grammars are **weakly equivalent** if they generate the same set of strings but do not assign the same phrase structure to each sentence

CHOMSKY NORMAL FORM

- We can define a normal form for grammars where each rule (production) has a particular form
- A CFG is in Chomsky normal form (CNF) if
  - It is ϵ-free (epsilon stands for the empty string)
  - Each rule is in the form $A \rightarrow B\ C$ or $A \rightarrow a$

  (The right-hand side of the rule can have two non-terminal symbols or 1 terminal symbol)
- CNF grammars are binary branching (they have binary trees)
- Any CFG can be converted into a weakly equivalent Chomsky normal form grammar
  - For example, a rule of the form $A \rightarrow B\ C\ D$ can be covered to the following two rules:

    $A \rightarrow B\ X$

    $X \rightarrow C\ D$
- A rule of the form $A \rightarrow A\ B$ generates a sequence starting with A, followed by an infinite sequence of B's is called a Chomsky-adjunction

CONSTITUENCY PARSING

CONSTITUENCY PARSING

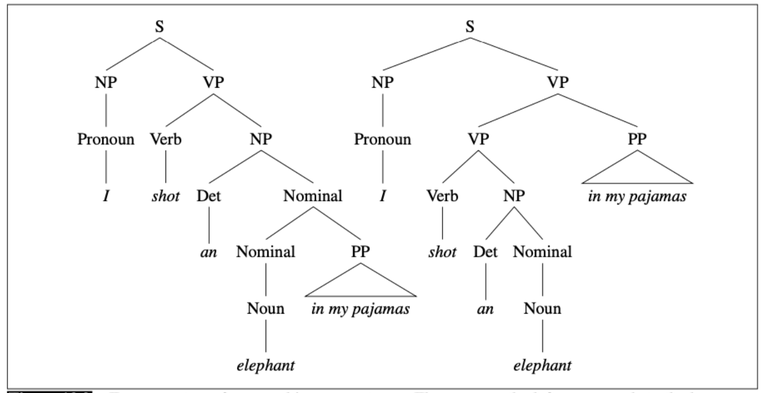


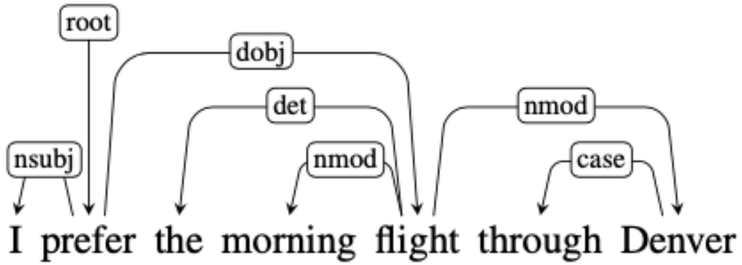
Figure 13.2 Two parse trees for an ambiguous sentence. The parse on the left corresponds to the humorous reading in which the elephant is in the pajamas, the parse on the right corresponds to the reading in which Captain Spaulding did the shooting in his pajamas.

- Syntactic parsing is the process of assigning syntactic structure to a sentence (based on a particular grammar)
- Constituency parsing is the process of assigning a parse structure (or tree) by a CFG
- Parse trees are useful for
  - Grammar checking (if a sentence cannot be parsed, it is likely incorrect or at least hard to read)
  - Question-Answering, e.g., in the sentence "Which flights to Denver depart before the Seattle flight?", knowing that "which flights to Denver" is the subject of "depart" helps us figure out that we are looking for flights from Denver to Seattle.
- Challenges
  - Structural ambiguity (a grammar may assign more than one parse to a sentence)
  - There are many grammatically correct but semantically unreasonable parses

DEPENDENCY PARSING

DEPENDENCY PARSING

- In the dependency formalism, the syntactic structure of a sentence is described in terms of the directed binary grammatical relations between (pairs of) words
  - Unlike constituent-based grammars and constituency parsing, constituents and phrase structure do not play a direct role in the dependency-based approach
- We have a fixed inventory of grammatical relations that are used to label arcs from heads to dependents (typed dependency structure)
- The root of the tree is the head of the entire structure
- The tree does not have phrasal constituents or lexical categories (like the constituent tree does)
- Dependency grammars are better at dealing with languages that are morphologically rich and have a more flexible word order (free word order)
  - In morphologically rich languages, each verb may have tens of forms based on tense, number, gender, etc.
- Dependency grammars skip the phrase analysis step and just focus on the "important relationships" directly



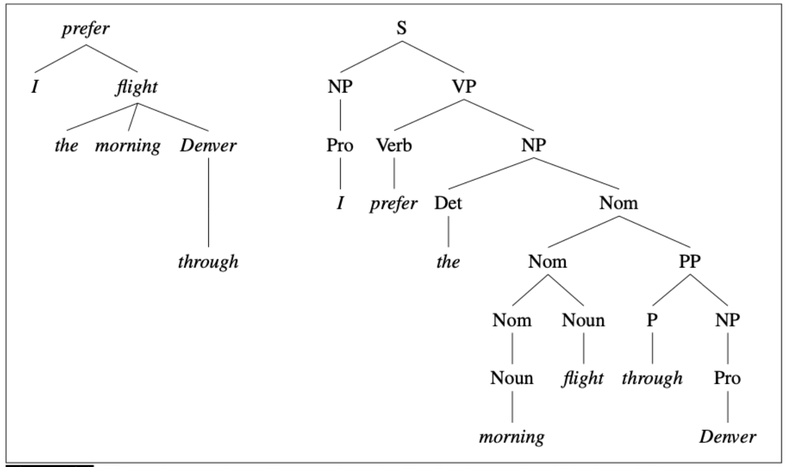
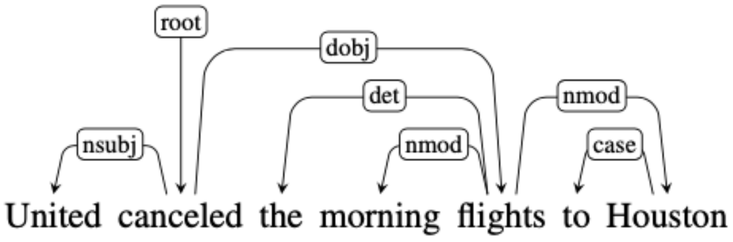


Figure 14.1 Dependency and constituent analyses for *I prefer the morning flight through Denver.*

DEPENDENCY PARSING (CONT.)

- In dependency parsing we have binary grammatical relations between a head and a dependent
- The Universal Dependencies (UD) project has assembled an inventory of dependency relations that are cross-linguistically applicable
  - Causal relations that describe syntactic roles with respect to a predicate (usually a verb)
    - Predicate is the part of a sentence containing a verb and stating something about the subject (e.g., "went home" in the sentence "Maria went home")
  - Modifier relations that categorize how words modify their heads
- In the example below,
  - The NSUBJ and DOBJ relations identify the subject and direct object of the predicate *cancel*
  - The NMOD, DET, and CASE denote modifiers of the noun's *flights* and *Houston*



Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 14.2 Some of the Universal Dependency relations (de Marneffe et al., 2014).

Relation	Examples with *head* and **dependent**
NSUBJ	**United** *canceled* the flight.
DOBJ	United *diverted* the **flight** to Reno.
	We *booked* her the first **flight** to Miami.
IOBJ	We *booked* **her** the flight to Miami.
NMOD	We took the **morning** *flight*.
AMOD	Book the **cheapest** *flight*.
NUMMOD	Before the storm JetBlue canceled **1000** *flights*.
APPOS	*United*, a **unit** of UAL, matched the fares.
DET	**The** *flight* was canceled.
	Which *flight* was delayed?
CONJ	We *flew* to Denver and **drove** to Steamboat.
CC	We flew to Denver **and** *drove* to Steamboat.
CASE	Book the flight **through** *Houston*.

Figure 14.3 Examples of core Universal Dependency relations.