Introduction to Phrase-Structure Parsing

Miguel Ballesteros

Algorithms for NLP Course.
7-11

Using some of Chris's last year slides in 711

Plan

- Parsing as Logical Deduction.
- Defining the CFG recognition problem.
- Bottom up vs. top down.

Algorithms for CFGs

- Given a CFG G and a string S:

  Recognition: Is $S \in L(G)$?

  - Equivalently, find some derivation that proves S is in G's language.

  Parsing:

  - What are (all of) G's derivations of S?
  - What is the "correct" derivation of S under G?

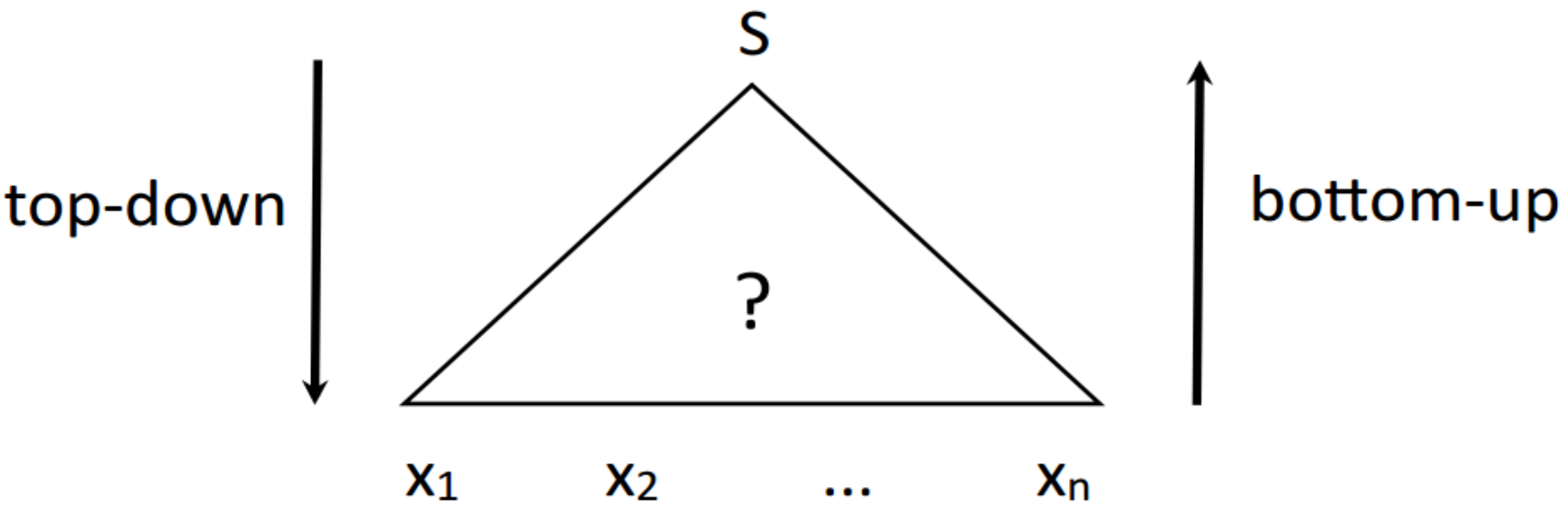
- The same core algorithms actually provide solutions to both!

Distinction

- Deterministic grammars give much faster recognition and parsing algorithms
  - For programming languages, they are parseable in linear time (compilers)
- For NLP, this is much slower.
- Today you will learn an $O(|N|^3 n^3)$ solution.

Parsing as search

- Top-down
- Bottom-up



Trees break into pieces (partial trees), which can be used to define a search space.

Top-Down Parsing (Recursive Descent)

- Input: “Book a flight”

Grammar	Lexicon
S → NP VP	*Det → that | this | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 13.1 The $\mathcal{L}_1$ miniature English grammar and lexicon.

Top-Down Parsing (Recursive Descent)

- Input: “Book a flight”

(S)

Grammar	Lexicon
S → NP VP	*Det → that | this | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 13.1 The $\mathcal{L}_1$ miniature English grammar and lexicon.

Top-Down Parsing (Recursive Descent)

- Input: “Book a flight”

(S)

(S (NP (VP)) (S Aux (NP VP)) (S (VP))

(S (NP Pronoun) (VP)) (S NP ProperNoun) (VP)) (S (NP Det Nominal) (VP))

Grammar	Lexicon
S → NP VP	*Det → that | this | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 13.1 The $\mathcal{L}_1$ miniature English grammar and lexicon.

Top-Down Parsing (Recursive Descent)

- Input: “Book a flight”

(S)

(S (NP (VP)) **(S Aux (NP VP))** (S (VP))

(S (NP Pronoun) (VP)) (S NP ProperNoun) (VP)) (S (NP Det Nominal) (VP))
(S Aux (NP Pronoun) (VP)) (S Aux (NP ProperNoun)(VP)) (S Aux (NP Det Nominal) (VP))

Grammar	Lexicon
S → NP VP	*Det → that | this | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 13.1 The $\mathcal{L}_1$ miniature English grammar and lexicon.

Top-Down Parsing (Recursive Descent)

- Input: “Book a flight”

(S)

(S (NP (VP)) (S Aux (NP VP)) **(S (VP))**

(S (NP Pronoun) (VP)) (S NP ProperNoun) (VP)) (S (NP Det Nominal) (VP))
(S Aux (NP Pronoun) (VP)) (S Aux (NP ProperNoun)(VP)) (S Aux (NP Det Nominal) (VP))
(S (VP (VP) (PP))) (S (VP Verb)) (S (VP Verb (NP))) (S (VP Verb (NP) (PP))) (S (VP Verb (PP)))

Grammar	Lexicon
S → NP VP	*Det → that | this | a*
S → Aux NP VP	*Noun → book | flight | meal | money*
S → VP	*Verb → book | include | prefer*
NP → Pronoun	*Pronoun → I | she | me*
NP → Proper-Noun	*Proper-Noun → Houston | NWA*
NP → Det Nominal	*Aux → does*
Nominal → Noun	*Preposition → from | to | on | near | through*
Nominal → Nominal Noun	
Nominal → Nominal PP	
VP → Verb	
VP → Verb NP	
VP → Verb NP PP	
VP → Verb PP	
VP → VP PP	
PP → Preposition NP	

Figure 13.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

Top-Down Parsing (Recursive Descent)

- Does not waste time exploring ungrammatical trees.
- Most search states (partial trees) will never lead to a derivation of our sentence, though.

- Left recursion problem …

Top-Down Parsing (Recursive Descent)

Left recursion problem:

(S)

(S (VP))

(S (VP (VP) (PP)))

(S (VP (VP (VP) (PP)) (PP)))

(S (VP (VP (VP (VP) (PP)) (PP)) (PP)))

(S (VP (VP (VP (VP (VP) (PP)) (PP)) (PP)) (PP)))

(S (VP (VP (VP (VP (VP (VP) (PP)) (PP)) (PP)) (PP)) (PP)))

...

Top-Down Recognition (Recursive Descent)

- Don't need to store the tree...
- Can collapse states that has the same functionality.
- Store unexpanded nonterminals (in sequence) only, along with the number of words “covered” so far.
  - Reminds to as generating from a CFG.

Bottom-Up Parsing

(Noun book) (Det that) (Noun flight) (Verb book) (Det that) (Noun flight)

Book that flight

Bottom-Up Parsing

(Nominal (Noun book)) (Det that) (Nominal (Noun flight))

(Noun book) (Det that) (Noun flight) (Verb book) (Det that) (Noun flight)

Book that flight

Bottom-Up Parsing

(Verb book) (Det that) (Nominal (Noun flight))

(Nominal (Noun book)) (Det that) (Nominal (Noun flight))

(Noun book) (Det that) (Noun flight) **(Verb book) (Det that) (Noun flight)**

Book that flight

Bottom-Up Parsing

(Nominal (Noun book)) (NP (Det that) (Nominal (Noun flight)))

(Verb book) (Det that) (Nominal (Noun flight))

(Nominal (Noun book)) (Det that) (Nominal (Noun flight))

(Noun book) (Det that) (Noun flight) (Verb book) (Det that) (Noun flight)

Book that flight

Bottom-Up Parsing

...

(Nominal (Noun book)) (NP (Det that) (Nominal (Noun flight)))

(Verb book) (Det that) (Nominal (Noun flight))

(Nominal (Noun book)) (Det that) (Nominal (Noun flight))

(Noun book) (Det that) (Noun flight) (Verb book) (Det that) (Noun flight)

Book that flight

Bottom-Up Parsing

- Never generates trees that are inconsistent with the sentence.
- Generates (lots of) partial trees that have no hope of getting to **S**.

Shift-Reduce

- A stack and a queue (or buffer).
- Remember PDAs?
  - Very similar.
- You are in a state, you can either SHIFT (PUSH to the stack) or Reduce (Pop from the stack)

- It is like a bottom up method.

Ambiguity

- A sentence may have many parses.
- Even if a sentence has only one parse, finding
- it may be difficult, because there are many misleading paths you could follow.
  - Bottom-up: fragments that can never have a home in any **S**.
  - Top-down: fragments that never get you to **x**

- What to do when there are many parses...
- how to choose? Return them all?