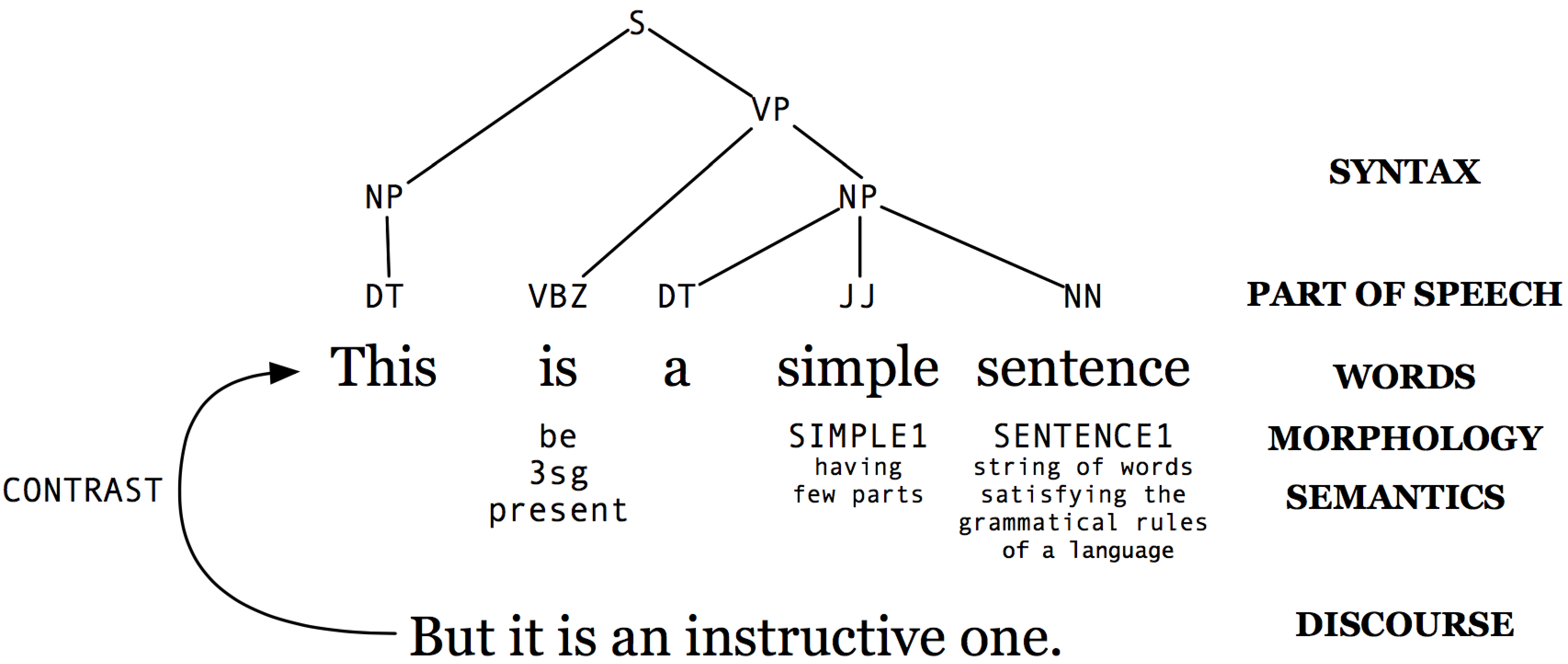
COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Syntax, Grammars & Parsing

CMSC 470

Marine Carpuat

Fig credits: Joakim Nivre, Dan Jurafsky & James Martin


S
VP
NP
NP
DT
VBZ
DT
JJ
NN
This is a simple sentence
be
3sg
present
SIMPLE1
having
few parts
SENTENCE1
string of words
satisfying the
grammatical rules
of a language
CONTRAST
But it is an instructive one.
SYNTAX
PART OF SPEECH
WORDS
MORPHOLOGY
SEMANTICS
DISCOURSE

Syntax & Grammar

- Syntax
  - From Greek syntaxis, meaning “setting out together”
  - refers to the way words are arranged together.

- Grammar
  - Set of structural rules governing composition of clauses, phrases, and words in any given natural language
  - Descriptive, not prescriptive
  - Panini’s grammar of Sanskrit ~2000 years ago

Syntax and Grammar

- Goal of syntactic theory
  - “explain how people combine words to form sentences and how children attain knowledge of sentence structure”

- Grammar
  - implicit knowledge of a native speaker
  - acquired without explicit instruction
  - minimally able to generate all and only the possible sentences of the language

[Philips, 2003]

Syntax in NLP

- Syntactic analysis can be useful in many NLP applications
  - Grammar checkers
  - Dialogue systems
  - Question answering
  - Information extraction
  - Machine translation
  - …

- Sequence models can go a long way but syntactic analysis is particularly useful
  - In low resource settings
  - In tasks where precise output structure matters

Two views of syntactic structure

- Constituency (phrase structure)
  - Phrase structure organizes words in nested constituents

- Dependency structure
  - Shows which words depend on (modify or are arguments of) which on other words

Constituency

- Basic idea: groups of words act as a single unit

- Constituents form coherent classes that behave similarly
  - With respect to their internal structure: e.g., at the core of a noun phrase is a noun
  - With respect to other constituents: e.g., noun phrases generally occur before verbs

Constituency: Example

- The following are all noun phrases in English...

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

- Why?
  - They can all precede verbs
  - They can all be preposed/postposed
  - …

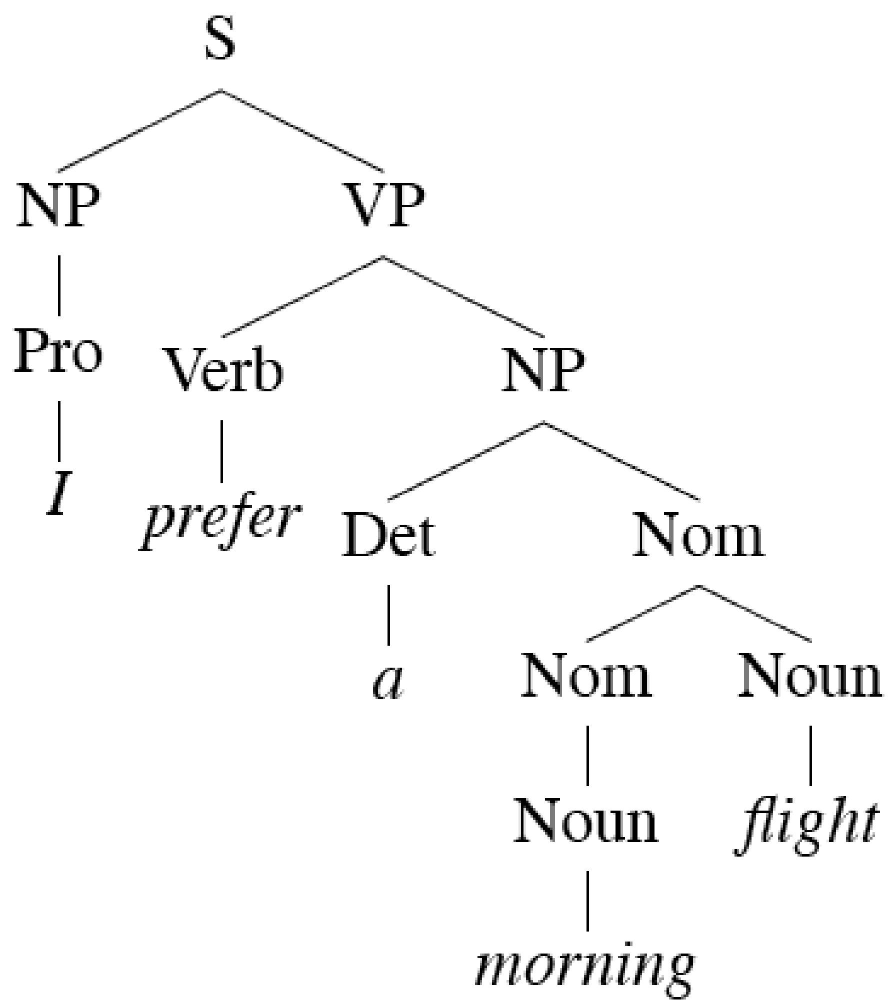
Grammars and Constituency

- For a particular language:
  - What are the “right” set of constituents?
  - What rules govern how they combine?
- Answer: not obvious and difficult
  - There are many different theories of grammar and competing analyses of the same data!

An Example Context-Free Grammar

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

Parse Tree: Example

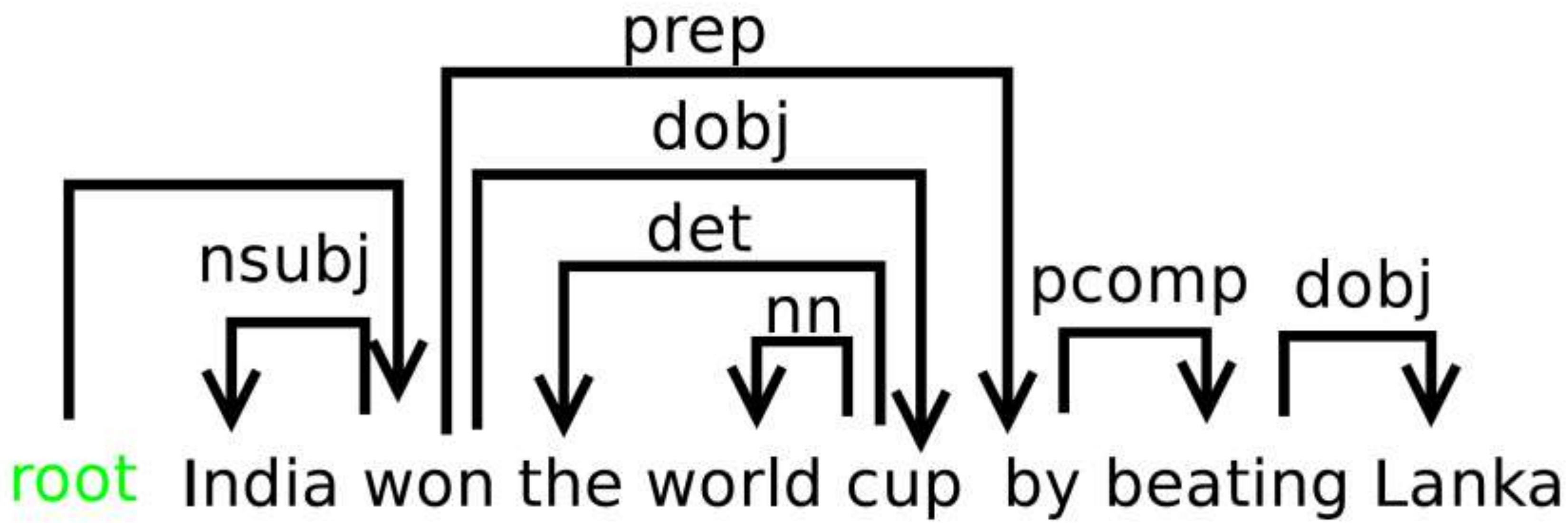


Dependency Grammars

- Context-Free Grammars focus on constituents
  - Non-terminals don’t actually appear in the sentence

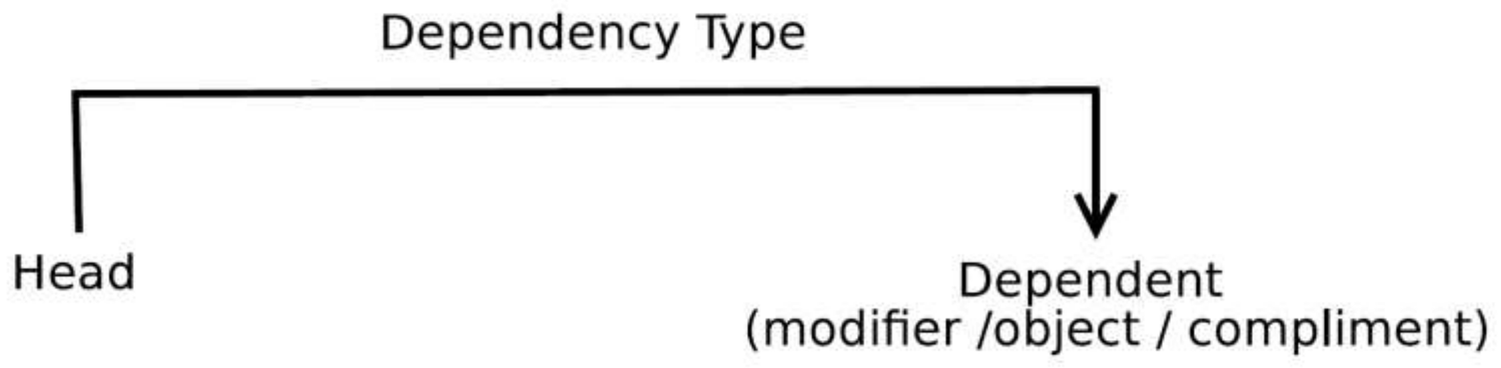
- In dependency grammar, a parse is a graph (usually a tree) where:
  - Nodes represent words
  - Edges represent dependency relations between words
    (typed or untyped, directed or undirected)

Example Dependency Parse



Dependency Grammars

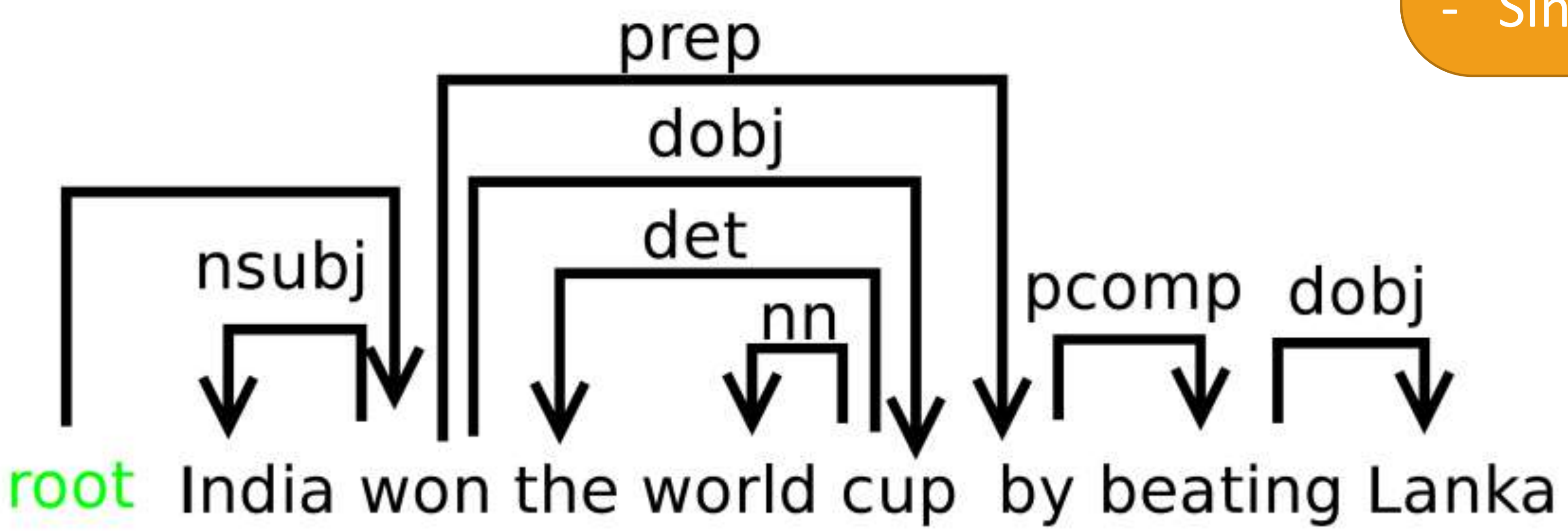
- Syntactic structure = lexical items linked by binary asymmetrical relations called dependencies



Example Dependency Parse

Dependencies form a tree:

- Connected
- Acyclic
- Single-head



Dependency Relations

Argument Dependencies	Description
nsubj	nominal subject
csubj	clausal subject
dobj	direct object
iobj	indirect object
pobj	object of preposition
Modifier Dependencies	**Description**
tmod	temporal modifier
appos	appositional modifier
det	determiner
prep	prepositional modifier

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.

Universal Dependencies project

- Set of dependency relations that are
  - Linguistically motivated
  - Computationally useful
  - Cross-linguistically applicable
  - [Nivre et al. 2016]

- Universaldependencies.org

Outline

- Syntax & Grammar

- Two views of syntactic structures
  - Context-Free Grammars
  - Dependency grammars
  - Can be used to capture various facts about the structure of language (but not all!)

- Dependency Parsing

Data-driven dependency parsing

Goal: learn a good predictor of dependency graphs

Input: sentence

Output: dependency graph/tree G = (V,A)

Can be framed as a structured prediction task

- very large output space
- with interdependent labels

2 dominant approaches: transition-based parsing and graph-based parsing

Transition-based dependency parsing

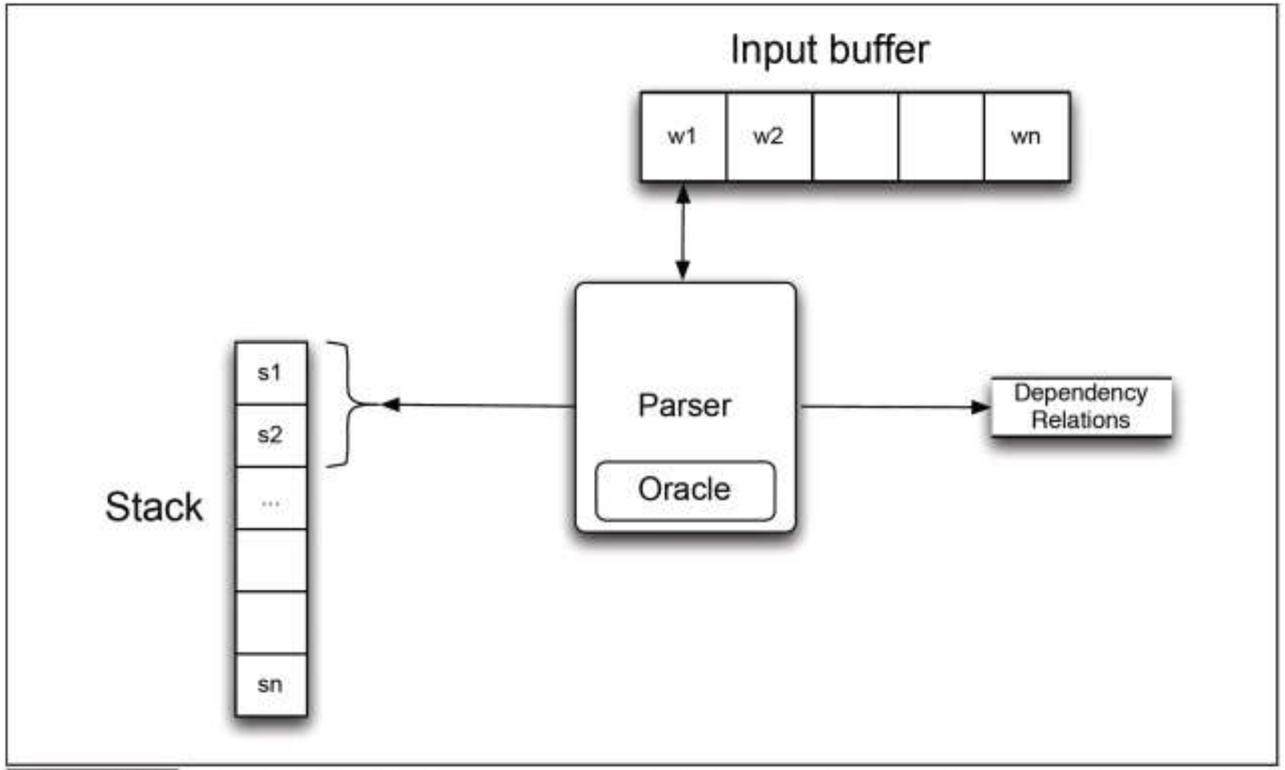


Figure 14.5 Basic transition-based parser. The parser examines the top two elements of the stack and selects an action based on consulting an oracle that examines the current configuration.

- Builds on shift-reduce parsing
  [Aho & Ullman, 1972]

- **Configuration**
  - **Stack**
  - **Input buffer** of words
  - Set of dependency relations

- Goal of parsing
  - find a final configuration where
  - all words accounted for
  - Relations form dependency tree

Defining Transitions

- **Transitions**
  - Are functions that produce a new configuration given current configuration
  - Parsing is the task of finding a sequence of transition that leads from start state to desired goal state
- **Start state**
  - Stack initialized with ROOT node
  - Input buffer initialized with words in sentence
  - Dependency relation set = empty
- **End state**
  - Stack and word lists are empty
  - Set of dependency relations = final parse

Arc Standard Transition System defines 3 transition operators [Covington, 2001; Nivre 2003]

LEFT-ARC

- create head-dependent relation between word at top of stack and 2^{nd} word (under top)
- remove 2^{nd} word from stack

RIGHT-ARC

- Create head-dependent relation between word on 2^{nd} word on stack and word on top
- Remove word at top of stack

SHIFT

- Remove word at head of input buffer
- Push it on the stack

Arc standard transition systems

- Preconditions
  - ROOT cannot have incoming arcs
  - LEFT-ARC cannot be applied when ROOT is the 2^{nd} element in stack
  - LEFT-ARC and RIGHT-ARC require 2 elements in stack to be applied

Transition-based Dependency Parser

function DEPENDENCYPARSE(*words*) **returns** dependency tree

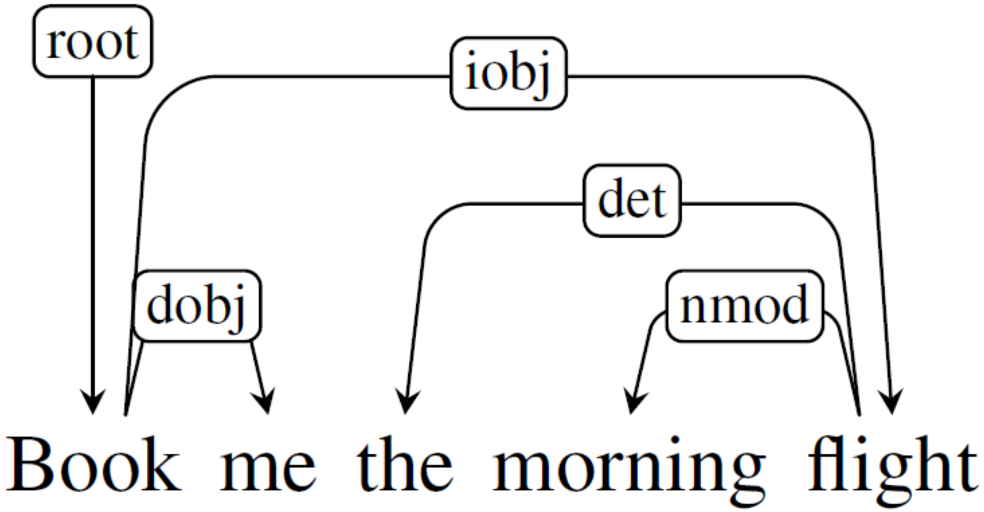
```
state ← {[root], [words], [] }  ; initial configuration
while state not final
    t ← ORACLE(state)           ; choose a transition operator to apply
    state ← APPLY(t, state)     ; apply it, creating a new state
return state
```

Figure 14.6 A generic transition-based dependency parser

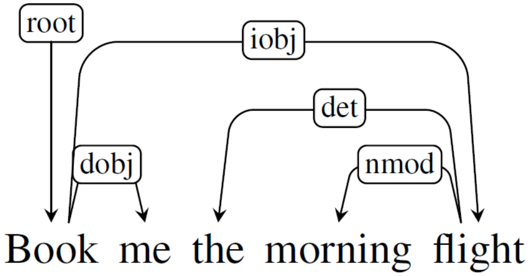
Properties of this algorithm:

- Linear in sentence length
- A greedy algorithm
- Output quality depends on oracle

Let’s parse this sentence



Transition-Based Parsing Illustrated



Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Figure 14.7 Trace of a transition-based parse.

Outline

- Syntax & Grammar

- Two views of syntactic structures
  - Context-Free Grammars
  - Dependency grammars
  - Can be used to capture various facts about the structure of language (but not all!)

- Dependency Parsing
  - Transition-based parser

Where do we get an oracle?

- Multiclass classification problem
  - Input: current parsing state (e.g., current and previous configurations)
  - Output: one transition among all possible transitions
  - Q: size of output space?

- Supervised classifiers can be used
  - E.g., perceptron
- Open questions
  - What are good features for this task?
  - Where do we get training examples?