NANYANG TECHNOLOGICAL UNIVERSITY

Natural Language Parsing: Formal Grammars

presented by

Jung-jae Kim

CSC421 Natural Language Processing

Review

Major Topics to cover

Methodology

- Word-level processing:
  - Regular expression, Spelling correction, segmentation
  - Language models, POS tagging
- Syntax: knowledge of structural relation between words
  - Parsing
- Semantics: Meaning
- Discourse: context

Applications:

- Information extraction, Named entity recognition
- Sentiment Analysis
- Machine translation

Syntax

- The way words are arranged together
  - Implicit knowledge of your native language that you had mastered by the time you were 3 years old without explicit instruction
  - Not the kind of stuff you were later taught in "grammar" school
- Why should we care?
  - Syntax is the key concept for many NLP tasks
    - E.g. relation identification
  - An approximation would be still useful

Key notions

- Constituency
- Grammatical relations
- Subcategorization

Constituent (1/2)

- Groups of words that behave as a single unit or phrase
  - Can all appear in similar syntactic environments
- E.g. Noun phrases (NPs) can occur before verbs
  - E.g. three parties from Brooklyn *arrive* …
    a high-class spot such as Mindy's *attracts* …
    the Broadway coppers *love* …
  - Cf. *from *arrive*
    *the *is*
    *spot *sat …*

Read Sections 12.2-3

- If you are NOT familiar with any of the following terms
  - Noun phrase (NP), verb phrase (VP), prepositional phrase (PP), adjective phrase
  - Declarative, imperative, clause
  - Determiner, nominal, quantifier
  - Auxiliaries, passive, perfect, progressive
  - Coordination, conjunction

Constituent (2/2)

- Evidence for constituency
  - Stick together through pre-/post-posed constructions
    - 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, I'd like to fly from Atlanta to Denver September seventeenth.
- Non-standard constructions (or constituents)
  - E.g. I gave John a watch and Mary a doll.
    *I gave John a watch Mary and a doll.

Grammatical relations

- Syntactic relations between constituents
  - Subject-verb
    - E.g. I love him.
  - Verb-object
    - E.g. I love him.
  - Verb-PrepositionalPhrase
    - E.g. I study at the library.
  - Noun-PrepositionalPhrase
    - E.g. The history of Singapore
  - Preposition-object

Verb phrases (VPs): Sample patterns

- Verb
  - disappear – *prefer
- Verb Object
  - prefer a morning flight – *disappear a morning flight
- Verb IObject DObject
  - give John a watch – *prefer John a watch
- Verb Object PrepositionalPhrase(PP)
  - leave Boston in the morning
- Verb PP
  - leave on Thursday

Subcategorization

- 'Subcategories' of different types of verbs
  - Verb
  - Verb NP
  - Verb NP NP
  - Verb NP PP
  - Verb PP
- Subcategorization frame
  - A possible set of complements for a verb
  - E.g. 'give': {}, {NP}, {NP, NP}, {NP, PP_{to}}, {PP}
  - Verb: predicate, complements: arguments

Exercise: Subcategorization

- Find all subcategorization frames for the verb ‘care’ based on the dictionary definitions

verb [no obj.]
1 [often with negative] feel concern or interest; attach importance to something: *they don't* ***care about*** *human life* | [with clause] *: I don't care what she says.*
• feel affection or liking: *you* ***care*** *very deeply* ***for*** *him.*
• (**care for something/care to do something**) like or be willing to do or have something: *would you care for some tea?* | *I don't care to listen to him.*
2 (**care for**) look after and provide for the needs of: *he has numerous animals to care for.*

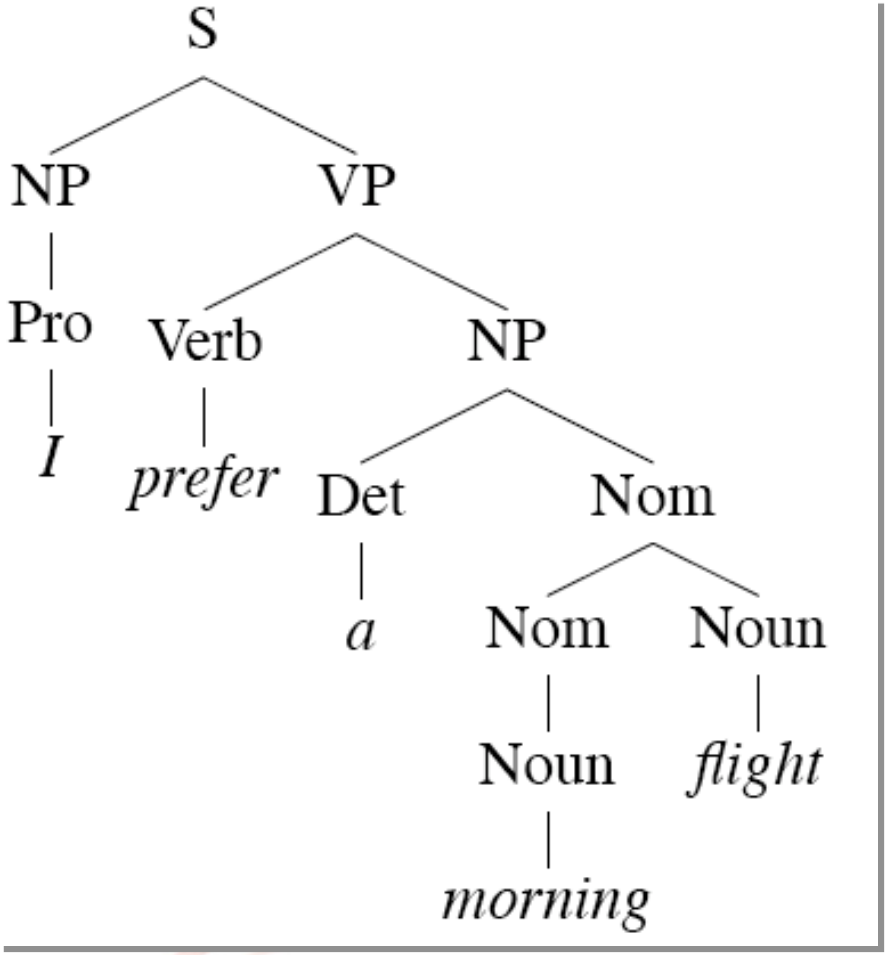
Syntactic structure

- Why do we learn about syntax in NLP course?
  - To identify and understand the syntactic structures of sentences
- Structural representation of grammatical relations expressed in a string
  - Or, called *parse tree*
- Representations
  - Phrase structure
  - Dependency structure
  - Predicate-argument structure

Phrase structure

- Organizes words into nested constituents
  - E.g. “I prefer a morning flight”

```
(S
  (NP (Pro I))
  (VP (Verb prefer)
    (NP (Det a)
      (Nom
        (Nom (Noun morning))
        (Noun flight)))))
```



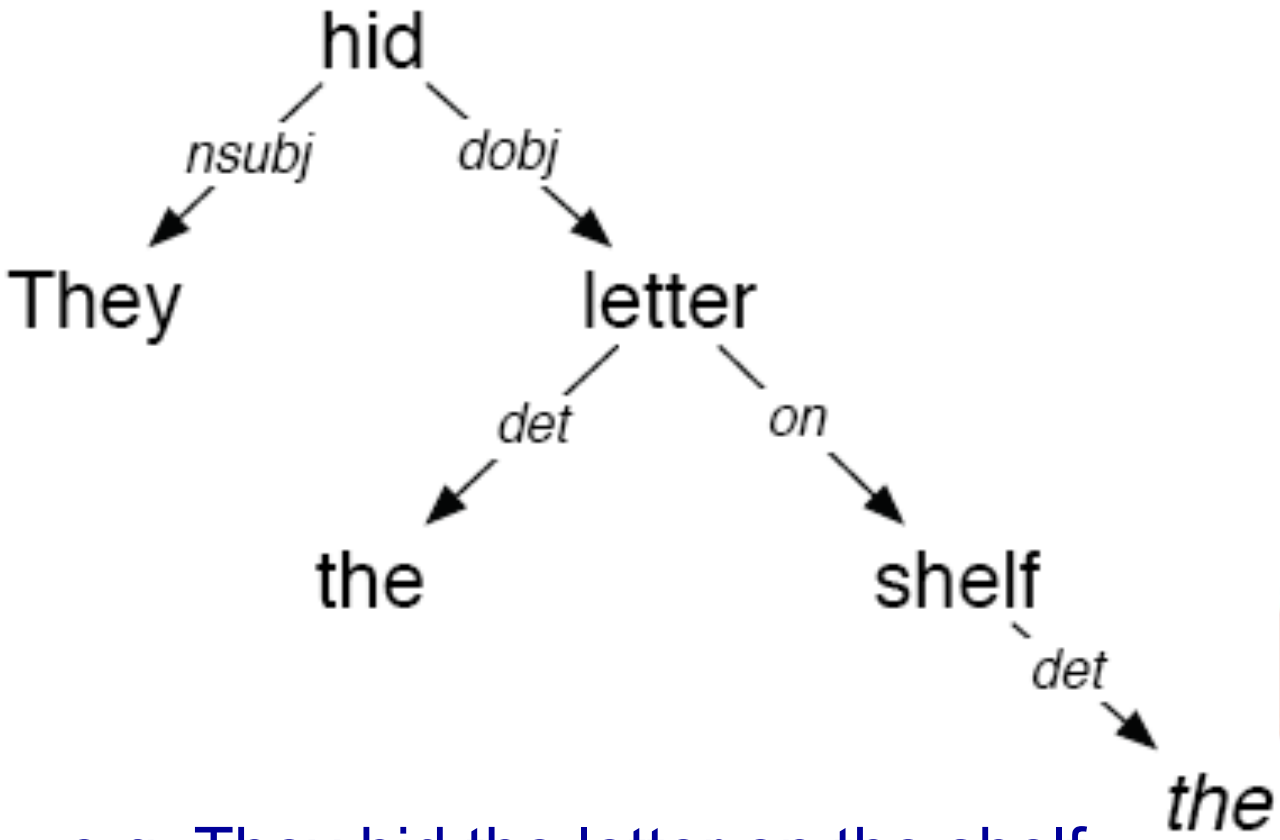
Read Section 12.3 for more syntactic tags

Exercise: Phrase structure

- Give the phrase structures of the two sentences:
  - (S1) John gives a present to Mary.
  - (S2) John gives Mary a present.
- POS tags:
  - E.g. ProperNoun: John, Mary
  - E.g. Verb, Det, Noun, Prep
- Example syntactic tags
  - E.g. S, NP, VP, PP_{XX}

Dependency structure

- Represents grammatical (dependency) relations between pairs of words (i.e. head, dependent)
- Head: Grammatically most important word in a phrase
  - Verb of VP
  - Noun of NP
  - Prep of PP
  - Adj of AdjP
  - …



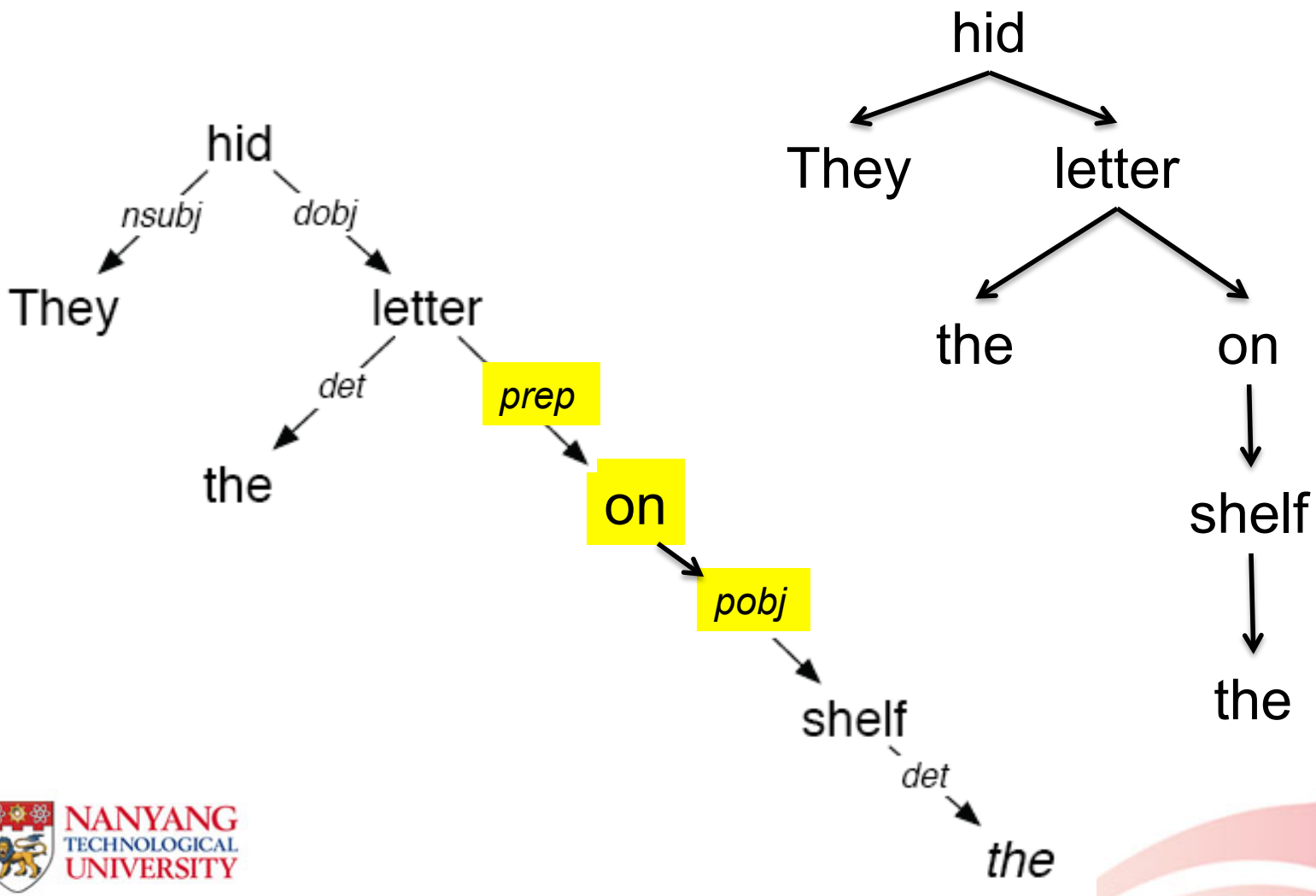
e.g. They hid the letter on the shelf

Dependency Relations

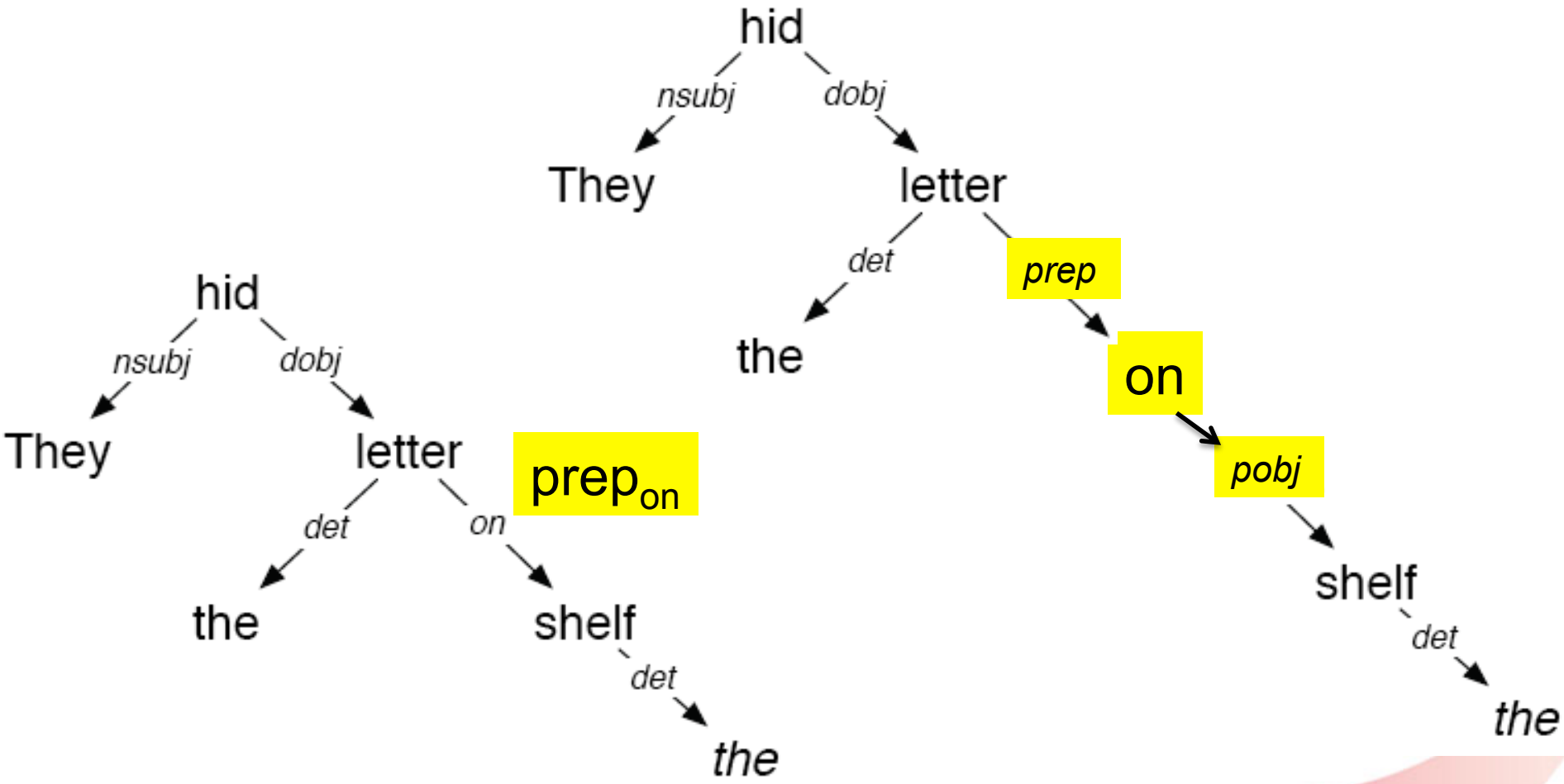
Argument Dependencies	Description
nsubj	nominal subject
csubj	clausal subject
dobj	direct object
iobj	indirect object
pobj	object of preposition
Modifier Dependencies	**Description**
tmod	temporal modifier
appos	appositional modifier
det	determiner
prep	prepositional modifier

Read Section 12.7 for more details

Typed dependency vs. untyped dependency



Typed dependency
vs. Collapsed typed dependency



Exercise: Dependency structure

- Give the dependency structures of the two sentences:
  - (S1) John gives a present to Mary.
  - (S2) John gives Mary a present.
- Example dependency relations
  - E.g. nsubj, iobj, dobj, det, $prep_{XX}$

Predicate-argument structure

- Review: Verb-predicate, complements-arguments
- Generalize to all types of words that have complements
  - E.g. Preposition-predicate, object-argument
- E.g. They hid the$_1$ letter on the$_2$ shelf.

Relation type	Predicate	Argument 1	Argument 2
verb_arg12	hid	They	letter
det_arg1	the$_1$	letter	
prep_arg12	on	letter	shelf
det_arg1	the$_2$	shelf	

Dependency structure vs. Predicate-argument structure

- e.g. Mary seems to like John.

Dependency structure

```
nsubj(seems-2, Mary-1)
root(ROOT-0, seems-2)
aux(like-4, to-3)
xcomp(seems-2, like-4)
dobj(like-4, John-5)
```

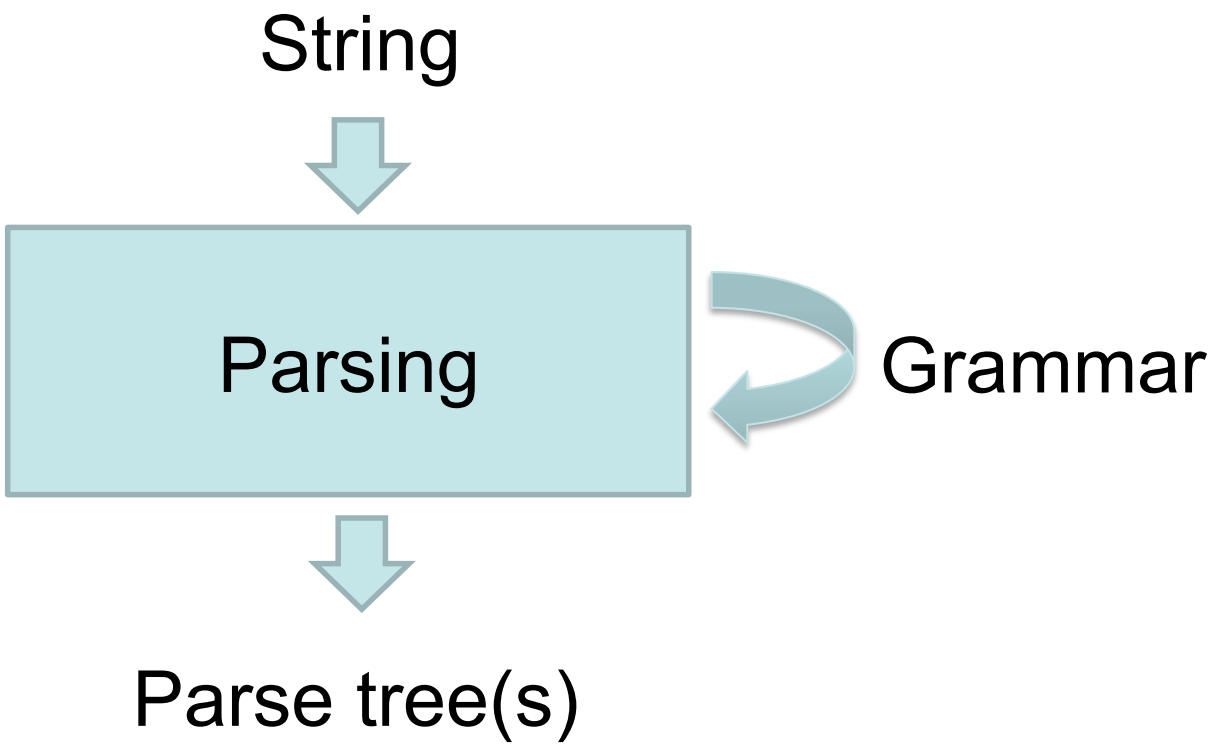
Predicate-argument structure

```
verb_arg12(seems-2, Mary-1, to-3)
comp_arg1(to-3, like-4)
verb_arg12(like-4, Mary-1, John-5)
```

Exercise: Predicate-argument structure

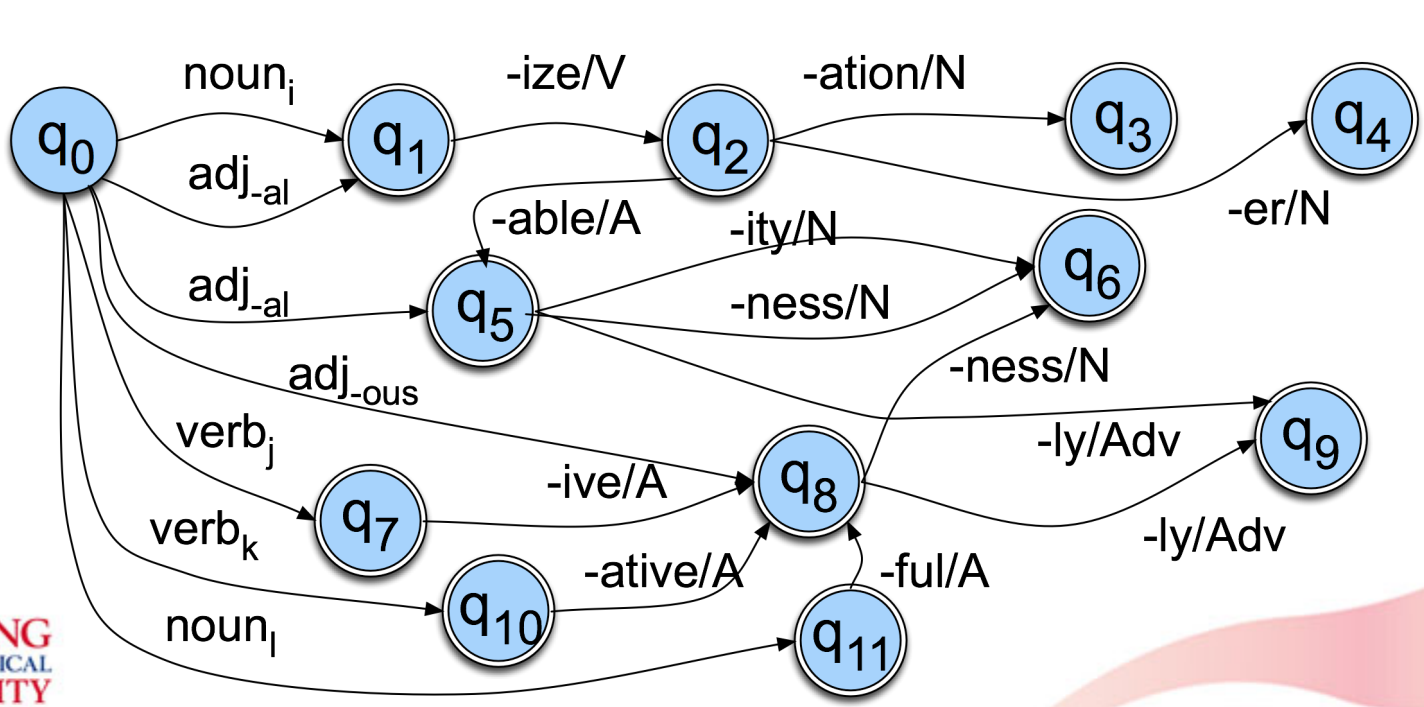
- Give the predicate-argument structure of the following sentences
  - John gives Mary a present
  - Mary is easy to please

Syntactic parsing



What is syntactic parsing?

- The process of taking a string and a grammar and returning parse tree(s) for that string
- It is completely analogous to running a finite-state transducer with a string

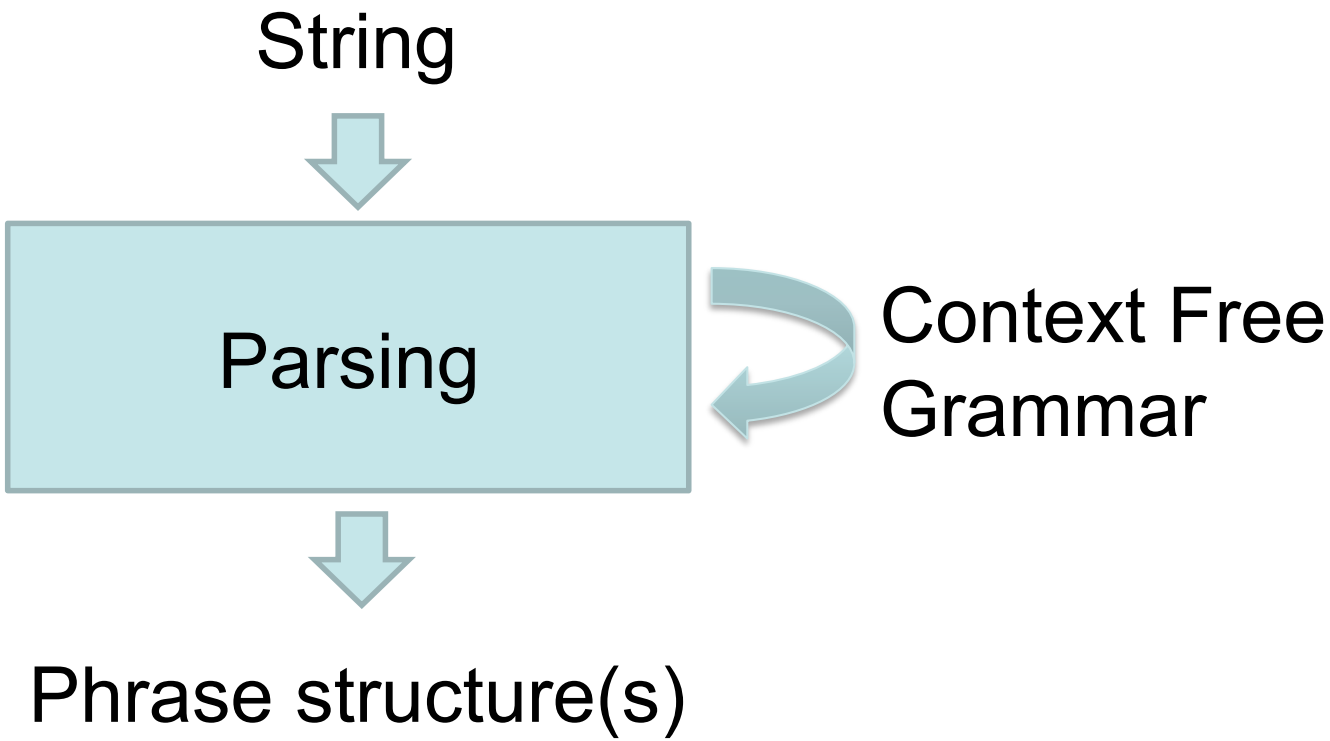


NANYANG TECHNOLOGICAL UNIVERSITY

Chomsky hierarchy

- But the grammar for parsing should be more powerful than finite-state transducer
- Chomsky hierarchy (Read Section 16.1 for details)
  - Regular Expressions (RE)
  - Context Free Grammars (CFG)
  - Context Sensitive Grammars (CSG)
  - Unrestricted Grammars (UG)

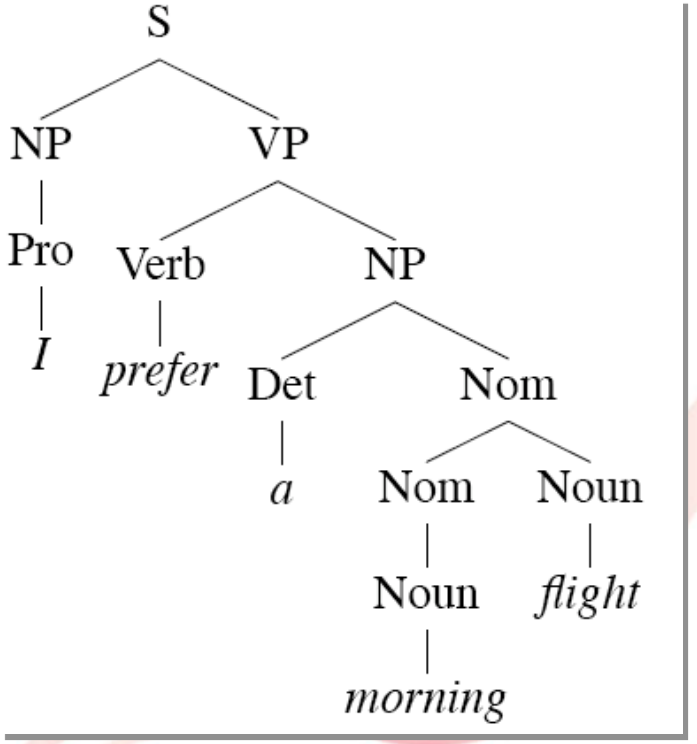
Syntactic parsing in the course



Context-free grammar (1/2)

- G = (T, N, S, R)
  - T: a set of terminals (e.g. ‘flight’)
  - N: a set of nonterminals (e.g. Noun)
  - S: the start symbol, a nonterminal
  - R: rules of the form X →γ
    - X: a nonterminal
    - γ: a sequence of terminals and nonterminals

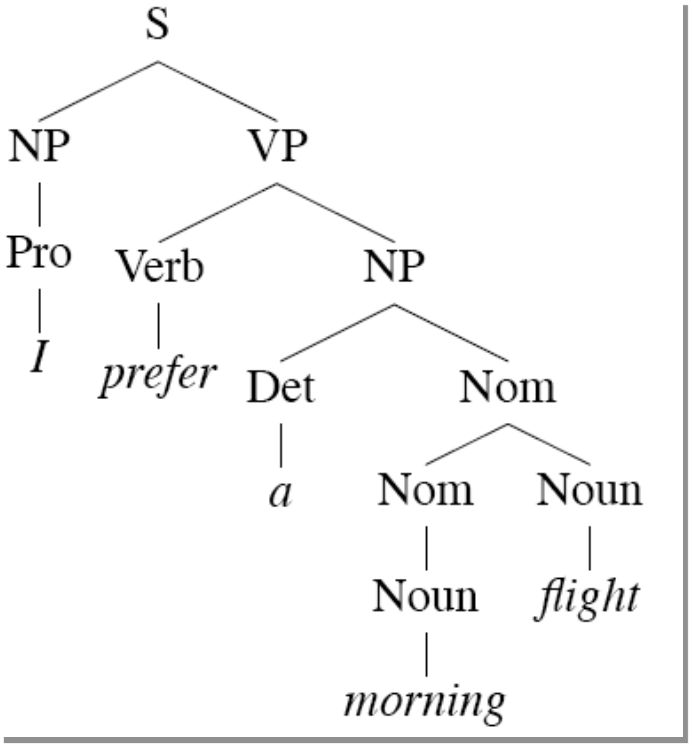
NP	→	*Det Nominal*
NP	→	*ProperNoun*
Nominal	→	*Noun* | *Nominal Noun*



Context-free grammar (2/2)

- G = (T, N, S, R)
  - T: words or tokens
  - N: POS tags, syntactic tags
  - S: the start symbol
  - R: rules of the form X →γ
    - X: a nonterminal
    - γ: the sequence of X's children

NP	→	*Det Nominal*
NP	→	*ProperNoun*
Nominal	→	*Noun* | *Nominal Noun*



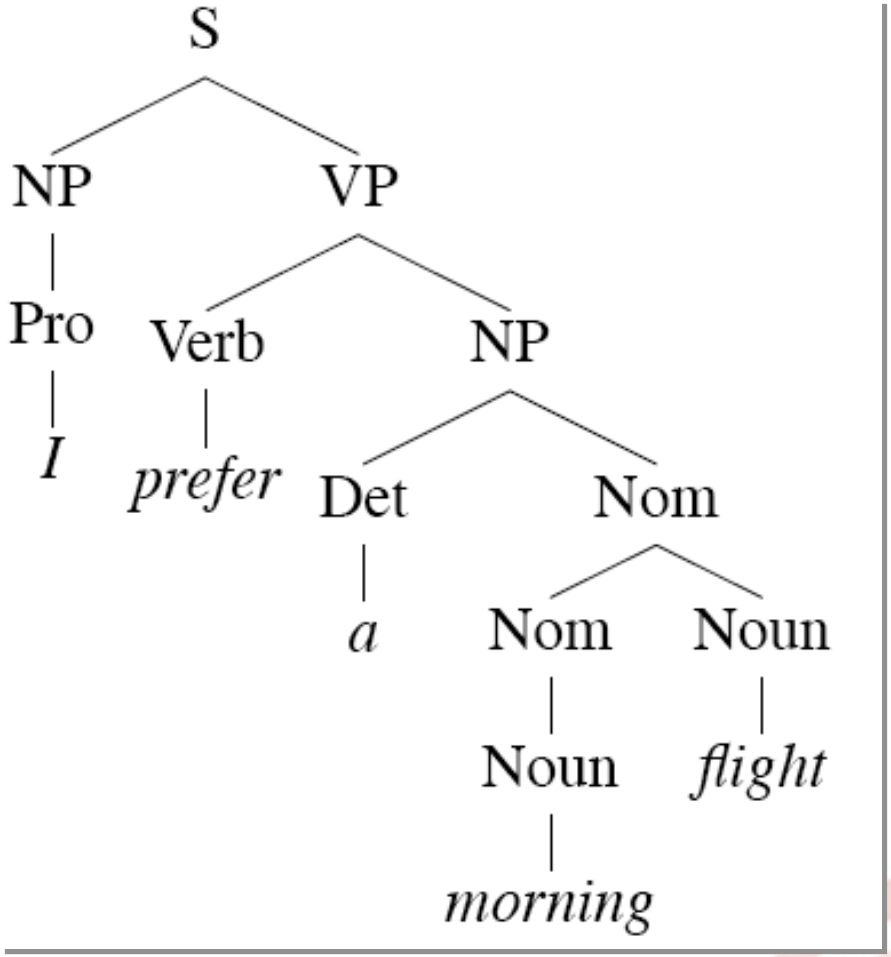
- A grammar G generates a language L
  - A language is a set of sentences

L0 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

Exercise: Context free grammar

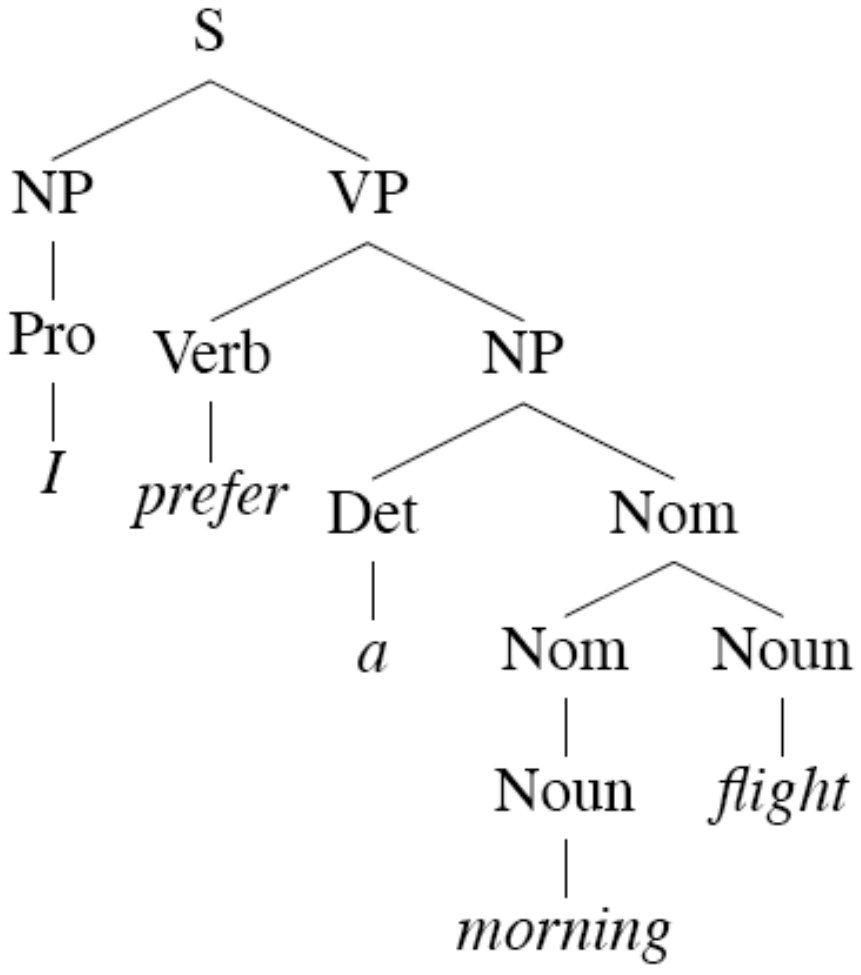
- Give a context free grammar to generate the parse tree



Derivation

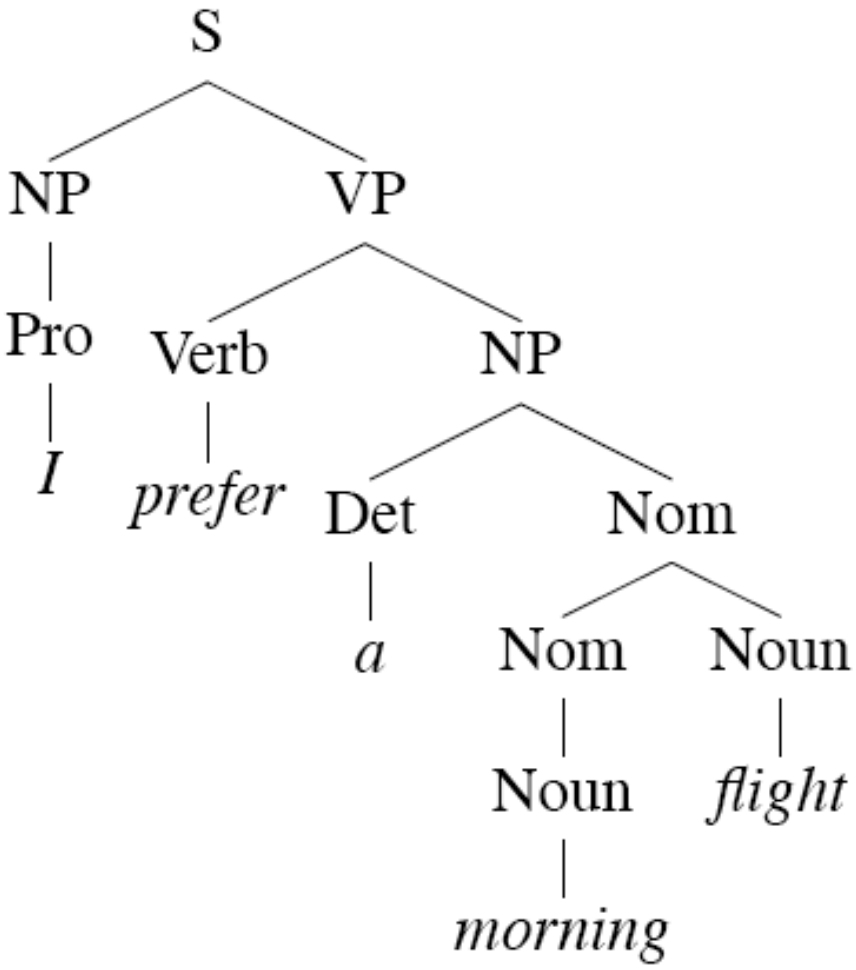
- A sequence of rules applied to a string
  - E.g. I prefer a morning flight.

$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Pronoun \\
&\quad\ |\ Proper\text{-}Noun \\
&\quad\ |\ Det\ Nominal \\
Nominal &\rightarrow Nominal\ Noun \\
&\quad\ |\ Noun \\
VP &\rightarrow Verb \\
&\quad\ |\ Verb\ NP \\
&\quad\ |\ Verb\ NP\ PP \\
&\quad\ |\ Verb\ PP \\
PP &\rightarrow Preposition\ NP
\end{aligned}
$$



Derivation (2)

- A sequence of rules applied to a string that *accounts* for that string



S → NP VP
NP → Pro
Pro → *I*
VP → Verb NP
Verb → *prefer*
NP → Det Nom
Det → *a*
Nom → Nom Noun
Nom → Noun
Noun → *morning*
Noun → *flight*

Summary: CFG & phrase structure

- G = (T, N, S, R)
  - Terminals: words
  - Non-terminals: constituent names (e.g. NP, VP)
  - Start symbol: S (sentence) or NP (noun phrase)
  - Rules: e.g. subcategorization frames for verbs
- Derivation will generate the phrase structure of an input

Example grammar formalisms

- Context free grammar (CFG)
- Dependency grammar (DG)
- Combinatory categorial grammar (CCG)

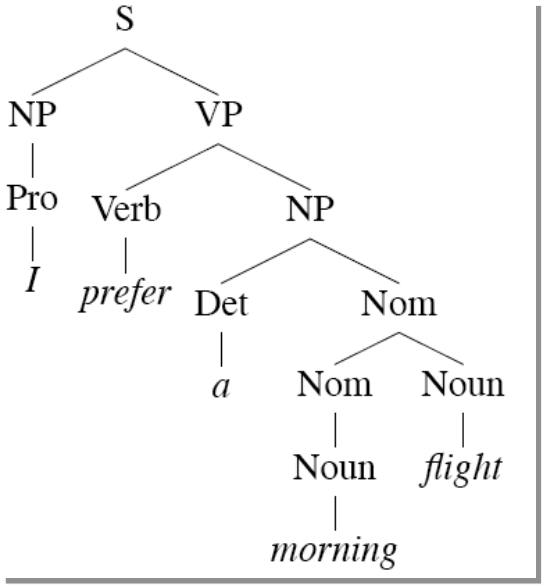
- HPSG: head-driven phrase structure grammar
- LFG: lexical functional grammar
- TAG: tree-adjoining grammar

Dependency Grammars

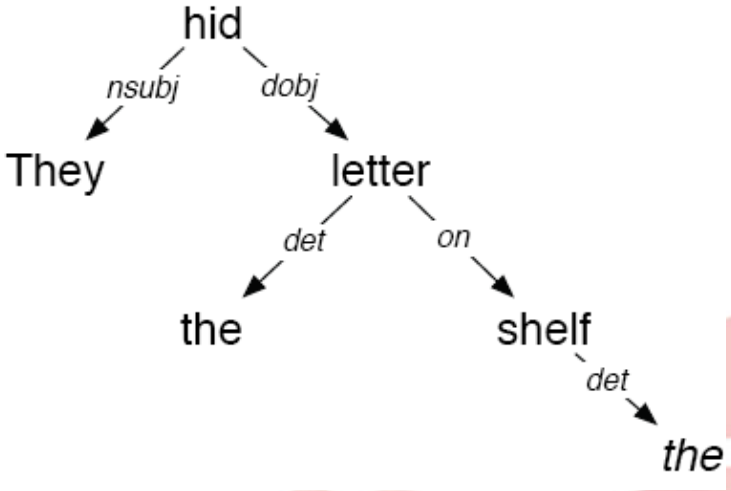
- In CFG-style phrase-structure grammars the main focus is on *constituents.*
- But it turns out you can get a lot done with just binary relations among the words in an utterance
  - Constituents are not required for many NLP tasks
- In a dependency grammar framework, a parse is a tree where
  - The links between the words represent dependency relations between pairs of words.
  - Relations may be typed (labeled), or not.

Phrase vs. dependency structure

- Constituents are specified
- Grammatical relations are implied
  - Can be deduced from rules



- Grammatical relations are specified
- A subtree corresponds to a constituent
  - Syntactic tag is implied by the root of subtree



Combinatory categorial grammar

- An extension of categorial grammar
- A lexicalized grammar model, consisting of
  - A categorial lexicon and
  - (a few) combinatory rules (or operators)
- Cf. CFG
  - POS tags for words (e.g. Noun, Verb)
  - (a lot of) rules (e.g. S → NP VP)
- Cf. dependency grammar
  - POS tags for words (N POS tags)
  - Dependency relations for all pairs of POS tags (N^2)
  - Annotated with probabilities

Categorial lexicon

- Operators
  - X/Y: a function that combines with a Y on its **right** to produce an X
  - X\Y: a function that combines with a Y on its **left** to produce an X
- Example categories
  - Determiner: NP/N (e.g. the boy)
  - Transitive verb: (S\NP)/NP (e.g. Harry eats apples)

Derivation of CCG

Harry	**eats**	**apples**	
NP	(S\NP)/NP	NP	
	S\NP		>
S			<

Application combinators

$$\frac{\alpha : X/Y \qquad \beta : Y}{\alpha\beta : X} >$$

$$\frac{\beta : Y \qquad \alpha : X\backslash Y}{\beta\alpha : X} <$$

NANYANG TECHNOLOGICAL UNIVERSITY

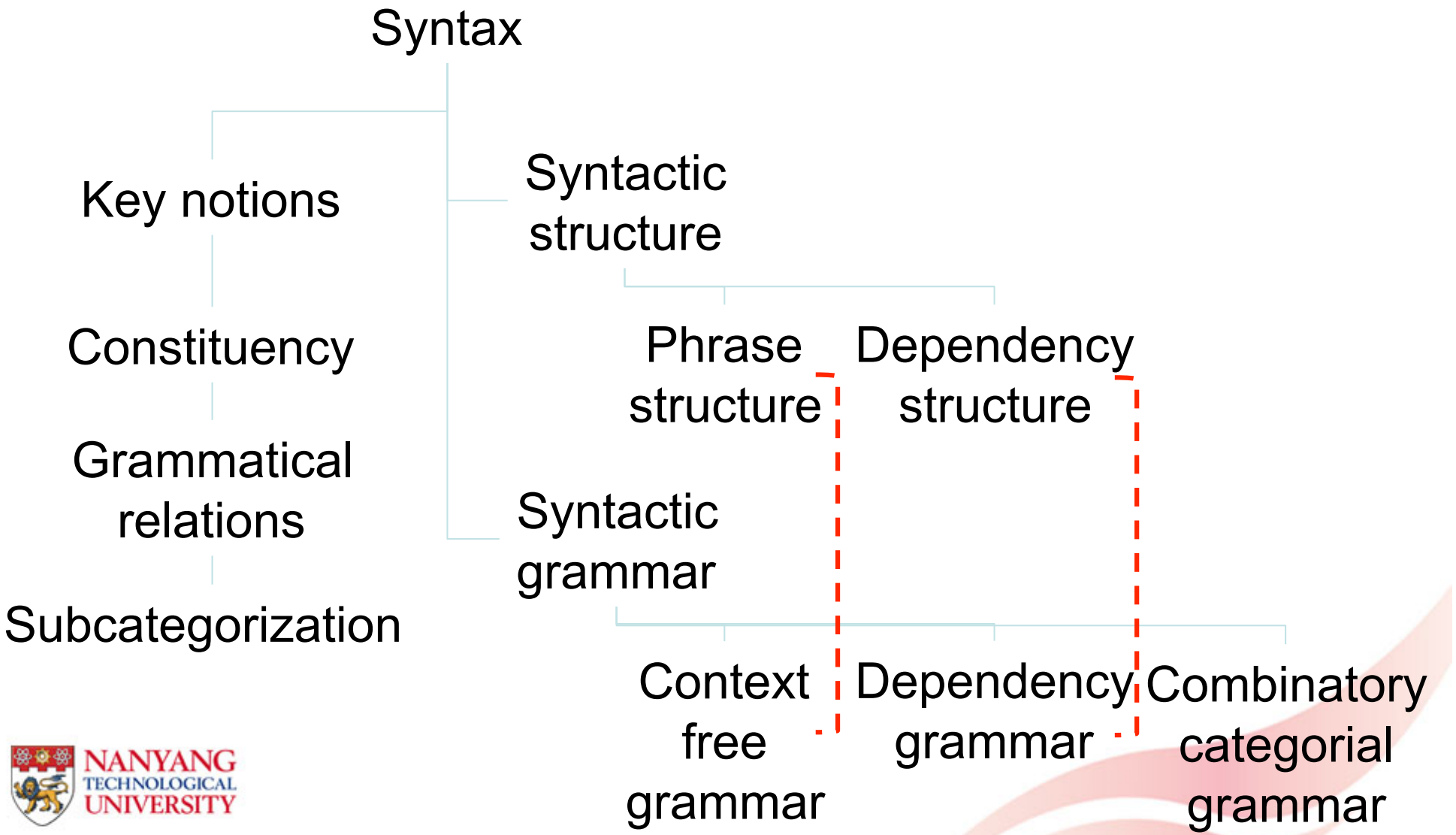
Exercise: CCG

- Give the derivation of the sentence “I prefer a morning flight”

We focus on CFG!

- Classic approach
  - Still widely used for practical parsing systems
- Theoretically well-studied
  - Equivalent to Backus-Naur Form (BNF)
  - SQL is formally defined in BNF

Summary



NANYANG TECHNOLOGICAL UNIVERSITY

Next class

Next class topics

- Syntactic parsing
- Strategies
  - Top-down parsing
  - Bottom-up parsing
- Algorithms
  - CKY parsing
  - Earley parsing
- Parse depth
  - Full parsing
  - Partial parsing

Reading assignments

- Chapter 13