Chart Parsing

Lecture #6

Computational Linguistics

CMPSCI 591N, Spring 2006

University of Massachusetts Amherst

Andrew McCallum

(Including slides from Jason Eisner)

Today's Main Points

- Hand back In-class Exercise #2
- Motivations and applications of Parsing.
- Dynamic Programming for Parsing: CYK
  – Some hands-on practice

- Discuss Programming Assignment #3
  "Implement CYK and build a grammar"

Programming languages

```
printf ("/charset [%s",
        (re_opcode_t) *(p - 1) == charset_not ? "^" : "");

assert (p + *p < pend);

for (c = 0; c < 256; c++)
  if (c / 8 < *p && (p[1 + (c/8)] & (1 << (c % 8)))) {
     /* Are we starting a range?  */
     if (last + 1 == c && ! inrange) {
        putchar ('-');
        inrange = 1;
     }
     /* Have we broken a range?  */
     else if (last + 1 != c && inrange) {
        putchar (last);
        inrange = 0;
     }

     if (! inrange)
        putchar (c);

     last = c;
  }
```

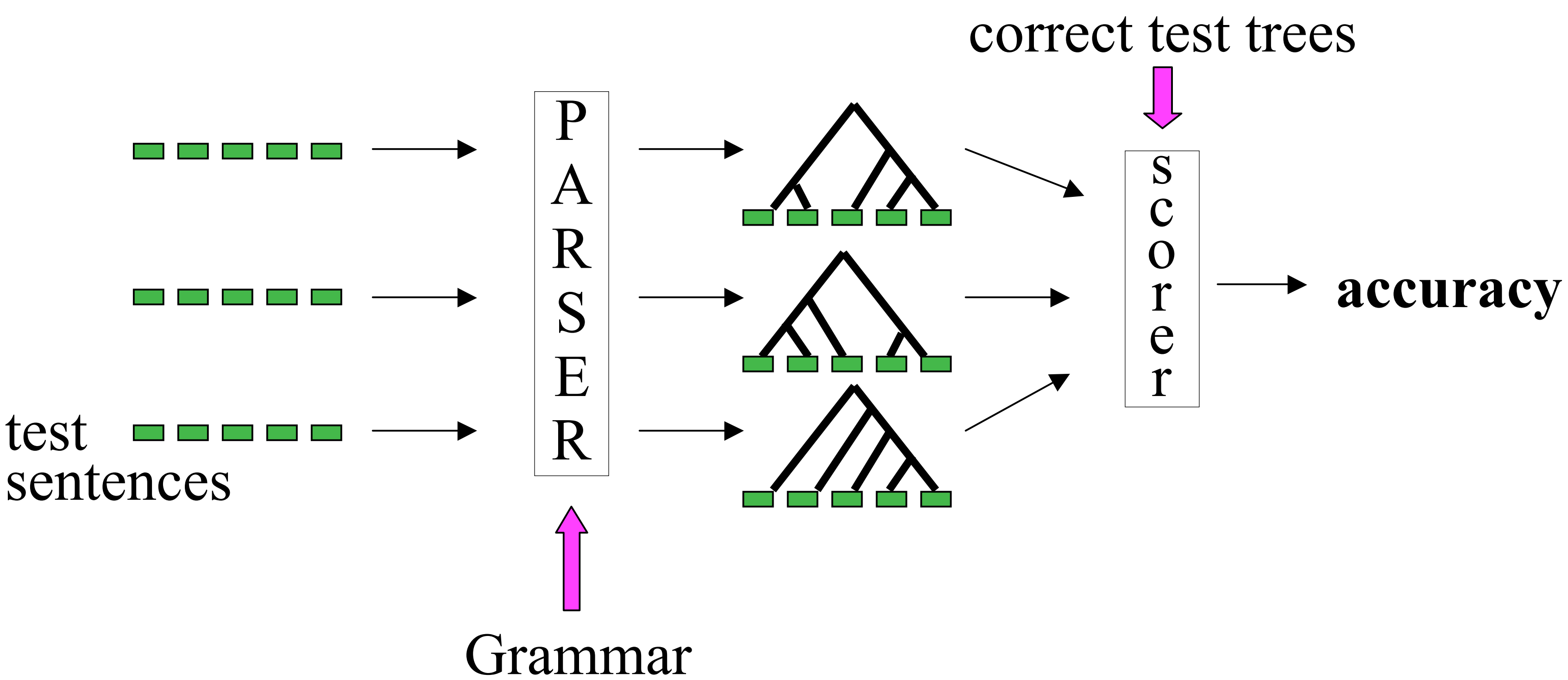
- Easy to parse.
- Designed that way!

Natural languages

```
printf "/charset %s", re_opcode_t *p - 1 == charset_not ? "^" : "";
assert p + *p < pend; for c = 0; c < 256; c++ if c / 8 < *p && p1 + c/8
& 1 << c % 8 Are we starting a range? if last + 1 == c && !  inrange
putchar '-'; inrange = 1; Have we broken a range? else if last + 1 != c
&& inrange putchar last; inrange = 0; if !  inrange putchar c; last =
c;
```

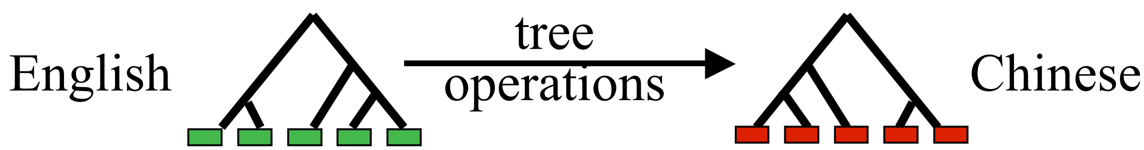
- No {} () [] to indicate scope & precedence
- Lots of overloading (arity varies)
- Grammar isn't known in advance!
- Context-free grammar not best formalism

The parsing problem



Applications of parsing (1/2)

- Machine translation (Alshawi 1996, Wu 1997, ...)



- Speech synthesis from parses (Prevost 1996)

  The government plans to raise income tax.

  The government plans to raise income tax the imagination.

- Speech recognition using parsing (Chelba et al 1998)

  Put the file in the folder.

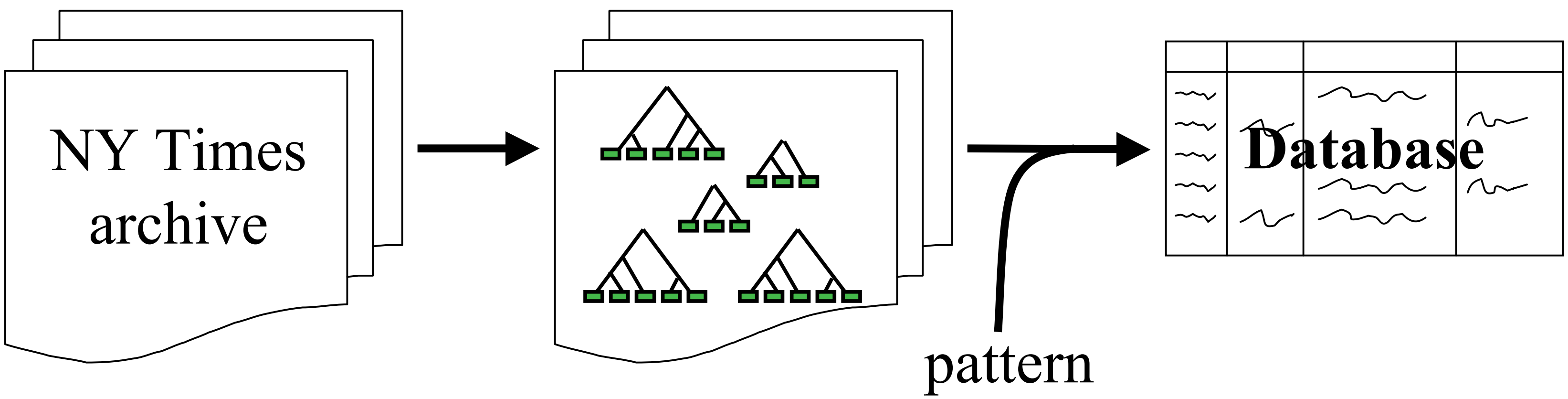
  Put the file and the folder.

Applications of parsing (2/2)

- Grammar checking (Microsoft)
- Indexing for information retrieval (Woods 1997)

  ... washing a car with a hose ... → vehicle maintenance

- Information extraction (Hobbs 1996) (Miller et al 2000)



Parsing State of the Art

- Recent parsers quite accurate, e.g.,
  - *A Maximum-Entropy-Inspired Parser*
    Eugene Charniak
    Proceedings of NAACL-2000.
  - *Three Generative, Lexicalised Models for Statistical Parsing*
    Michael Collins
    Proceedings of ACL, 1997.
- Most sentences parsed correctly, or with one error

Last class…

- We defined a CFG,
  where it sits in the Chomsky hierarchy
- Talked about parsing as ***search***…
  …through an exponential number of possible trees
- Gave examples of bottom-up and top-down search.
- Discussed problems:
  – Infinite loop with left-recursive rules
  – Much duplicated work in exponential space… backtracking

Dynamic Programming for Parsing

- Given CFG in Chomsky Normal Form, and an input string, we want to search for valid parse trees.
- What are the intermediate sub-problems?
- What would the dynamic programming table look like?

CKY algorithm, recognizer version

- Input: string of n words
- Output: yes/no (since it's only a recognizer)
- Data structure: n x n table
  - rows labeled 0 to n-1
  - columns labeled 1 to n
  - cell [i,j] lists possible constituents spanning words between i and j

CKY algorithm, recognizer version

- for i := 1 to n
  - Add to [i-1,i] all (part-of-speech) categories for the ith word
- for width := 2 to n
  - for start := 0 to n-width
    - Define end := start + width
    - for mid := start+1 to end-1
      - for every constituent X in [start,mid]
      - for every constituent Y in [mid,end]
      - for all ways of combining X and Y (if any)
      - Add the resulting constituent to [start,end] ~~if it's not already there.~~

	time 1	flies 2	like 3	an 4	arrow 5
0	**NP 3** **Vst 3**				
1		**NP 4** **VP 4**			
2			**P 2** **V 5**		
3				**Det 1**	
4					**N 8**

NP → time
Vst → time
NP → flies
VP → flies
P → like
V → like
Det → an
N → arrow

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3				
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4			
2			P 2 V 5		
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4	–		
2			P 2 V 5	–	
3				Det 1	
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4	–		
2			P 2 V 5	–	
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13			
1		NP 4 VP 4	–		
2			P 2 V 5	–	
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	**NP 3** **Vst 3**	**NP 10** **S 8** **S 13**	**–**		
1		**NP 4** **VP 4**	**–**	**–**	
2			**P 2** **V 5**	**–**	**PP 12**
3				**Det 1**	**NP 10**
4					**N 8**

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–		
1		NP 4 VP 4	–	–	
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

0	NP 3 Vst 3	NP 10 S 8 S 13	–		
1		NP 4 VP 4	–	–	
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

time 1 flies 2 like 3 an 4 arrow 5

	1	2	3	4	5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	
1		NP 4 VP 4	–	–	NP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

time **1** flies **2** like **3** an **4** arrow **5**

	1	2	3	4	5
0	**NP 3** **Vst 3**	**NP 10** **S 8** **S 13**	–	–	
1		**NP 4** **VP 4**	–	–	**NP 18** **S 21**
2			**P 2** **V 5**	–	**PP 12** **VP 16**
3				**Det 1**	**NP 10**
4					**N 8**

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1 flies 2	like 3	an 4	arrow 5	
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	NP 24
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	NP 24 S 22
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	NP 24 S 22 S 27
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24
1		NP 4 VP 4	__	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

time **1** flies **2** like **3** an **4** arrow **5**

0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27 S 22
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

Follow backpointers …

S

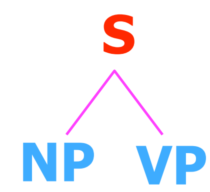
	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP



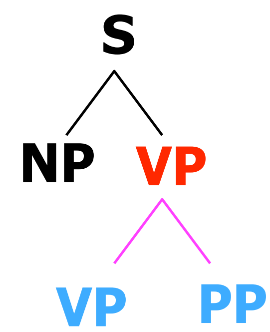
	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8

1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP



time	1 flies	2 like	3 an	4 arrow	5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8

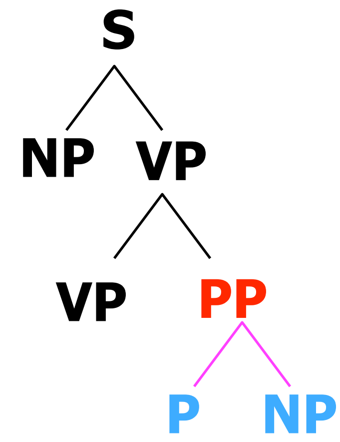
1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	–	–	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	–	–	NP 18 S 21 VP 18
2			P 2 V 5	–	PP 12 VP 16
3				Det 1	NP 10
4					N 8



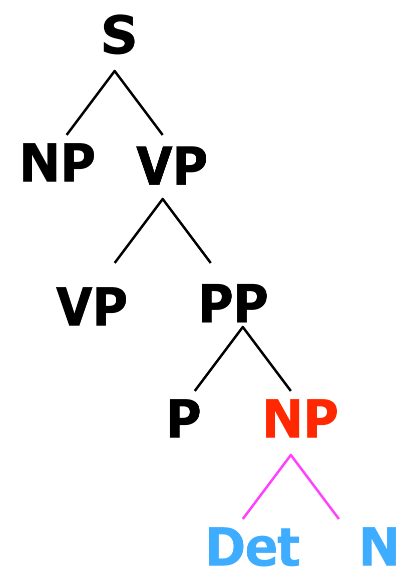
1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

	time 1	flies 2	like 3	an 4	arrow 5
0	NP 3 Vst 3	NP 10 S 8 S 13	_	_	NP 24 S 22 S 27 NP 24 S 27 S 22 S 27
1		NP 4 VP 4	_	_	NP 18 S 21 VP 18
2			P 2 V 5	_	PP 12 VP 16
3				Det 1	NP 10
4					N 8



1 S → NP VP
6 S → Vst NP
2 S → S PP

1 VP → V NP
2 VP → VP PP

1 NP → Det N
2 NP → NP PP
3 NP → NP NP

0 PP → P NP

CMPSCI 591N In-class Exercise #3 Name: ____________________ Student ID: _____________

Fill in the CYK dynamic programming table to parse the sentence below. In the bottom right corner, draw the two parse trees.

	she	eats	fish	with	chop-sticks
	0 1	2	3	4	5
0	NP				
1					
2					
3					
4					

- S → NP VP
- NP → NP PP
- VP → V NP
- VP → VP PP
- PP → P NP
- NP → she
- NP → fish
- NP → fork
- NP → chopsticks
- V → eats
- V → fish
- P → with

CMPSCI 591N In-class Exercise #3 Name: ____________________ Student ID: _____________

Fill in the CYK dynamic programming table to parse the sentence below. In the bottom right corner, draw the two parse trees.

she eats fish with chop-sticks

	1	2	3	4	5
0	NP		S NP,VP		S NP VP S NP VP
1		V	VP V NP		VP V NP VP VP PP
2			NP V		NP NP PP
3				P	PP P NP
4					NP

S → NP VP
NP → NP PP
VP → V NP
VP → VP PP
PP → P NP

NP → she
NP → fish
NP → fork
NP → chopsticks
V → eats
V → fish
P → with

Homework #3

- Implement CYK
- Create a grammar
- Experiment with it...

- No class next Tuesday.
- Homework still due Thursday.