School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2017)

Sample CYK solution: Week 4

Discussion

2. Consider the following simple **context–free grammar**:

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

(a) What changes need to be made to the grammar to make it suitable for **CYK parsing**?

- For CYK parsing, a grammar needs to be written in Chomsky Normal Form, where each rule consists of either:
  - a (single) non-terminal which re-writes as a single terminal, or
  - a (single) non-terminal which re-writes as exactly two non-terminals
- Here, we have two rules where a non-terminal re-writes as three non-terminals (`VP -> V NP PP` and `NP -> Det N PP`); we remove these rules and replace them with the following:

```
VP -> V X
X -> NP PP
NP -> Det Y
Y -> N PP
```

(b) Using the CYK strategy and the above grammar in CNF, parse the following sentences:

(ii) "an park by Bob walked an park with Bob"

- This is the most interesting example; we will work through it in detail.
- The basic chart is indicated in Table 1; we will need to examine all of the cells (marked ?) except for the ones along the bottom of each column, which are marked with the part(s)–of–speech for each token.
- We're going to work our way through the chart from left–to–right, bottom–to–top, filling in the ? cells with **partial parses**, according to what the grammar permits for various slices of the sentence. The cells labels indicate the **lexical span** of the slice. For example, [2,5] refers to the sentence fragment "by Bob walked"; if there are any non-terminals which can produce this sequence of tokens, according to the given grammar, we will record them in the cell labelled [2,5].

- For the leftmost column, there is only a single cell [0,1], marked with `Det`, because of the rule `Det->"an"`. If there were other parts–of–speech (**pre-terminals**) licensed for this token in the grammar (**lexical ambiguity**), then they would also be indicated in this cell.
- For the second column, (under *park*), we start from the bottom, where cell [1,2] is labelled with `N` (`N->"park"`). Cell [0,2] corresponds to the fragment "an park"; we attempt to find a parse for this fragment by considering the two neighbouring cells [0,1] and [1,2]. (Note that together, these combine to span [0,2].) [0,1] has the non-terminal `Det` and [1,2] has the non-terminal `N`. Is there a rule in our (CNF) grammar with `Det N` on the right-hand side? Yes: `NP->Det N`. We label this cell with `NP` (see Table 2).
- For the third column (under *by*), we again start from the bottom, where [2,3] is labelled with `P` (`P->"by"`). [1,3], corresponding to "park by", is an `N` ([1,2]) and a `P` ([2,3]). Is there a rule with `N P` on the right-hand side? No, so we leave this cell blank.
- For cell [0,3] ("an park by"), we want to consider every pair[1] of cells which together combine to span [0,3][2]. It turns out that there are two possible pairs:
  - [0,1] (`Det`) and [1,3] (blank): we know that `Det` is a non-terminal for [0,1] but we just determined that [1,3] is blank. Since there is no non-terminal that can generate [1,3], this combination cannot result in a parse for [0,3].
  - [0,2] (`NP`) and [2,3] (`P`): we determined that `NP` is the only possible non-terminal which covers the span [0,2]; is there a rule with `NP P`[3] on the right-hand side? No, so this cell is also left blank. (See Table 3.)
- For the fourth column (under *Bob*), we have an `NP` at [3,4]. At [2,4], we have `P NP` (from [2,3] and [3,4]): this is the right hand side of the rule `PP->P NP`, so we record `PP` here.
- At [1,4] ("park by Bob"), there are two possible combinations to form a span:
  - [1,2] and [2,4]: this comprises an `N` and a `PP`, which is the right hand side of the rule `Y->N PP`. We record `Y` in this cell.
  - [1,3] and [3,4]: [1,3] is blank, so there is no partial parse here; this doesn't affect the `Y` from the other combination, however.
- At [0,4] ("an park by Bob"), there are now three possible combinations:
  - [0,1] and [1,4]: `Det` and `Y`, which together make an `NP` (`NP->Det Y`).
  - [0,2] and [2,4]: `NP` and `PP`, which together can make an `X` (`X->NP PP`).
  - [0,3] and [3,4]: [0,3] is blank, so nothing.

[1]Note that this requirement, of only considering two cells at any given moment, allows us to find the parse(s) efficiently, and is the main reason that the grammar need to be in Chomsky Normal Form.

[2]Where the span is only of length 2, like for [0,2] and [1,3], there is only one such pair of cells, based on the immediate left and lower neighbour.

[3]Note that order matters; there is a rule with `P NP`, but we have "an park by" here, and not "by an park".

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] ?	[0,3] ?	[0,4] ?	[0,5] ?	[0,6] ?	[0,7] ?	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] ?	[1,4] ?	[1,5] ?	[1,6] ?	[1,7] ?	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] ?	[2,5] ?	[2,6] ?	[2,7] ?	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] ?	[3,6] ?	[3,7] ?	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] ?	[4,7] ?	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] ?	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 1: Initialised CYK chart

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] ?	[0,4] ?	[0,5] ?	[0,6] ?	[0,7] ?	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] ?	[1,4] ?	[1,5] ?	[1,6] ?	[1,7] ?	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] ?	[2,5] ?	[2,6] ?	[2,7] ?	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] ?	[3,6] ?	[3,7] ?	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] ?	[4,7] ?	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] ?	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 2: Cell [0,2] (spanning “an park”)

- Consequently, we add both `NP` and `X` to cell [0,4], because both of these two non-terminals can generate the sentence fragment "an park by Bob". (See Table 4.) We cannot determine, at this point, if either (or both) of them will contribute to the parse of the entire sentence, however.
- For the fifth column (under `walked`), we have a `V` at [4,5]. At [3,5], we consider `NP V` (from [3,4] and [4,5]), but there is no rule with that on the right-hand side, so we leave it blank.
- At [2,5], we consider [2,3] and [3,5] — but [3,5] is blank — and [2,4] [4,5], but `PP V` does not appear on the right-hand side of any rule, so it is also blank.
- At [1,5], we consider [1,2] and [2,5] — but [2,5] is blank — and [1,3] and [3,5] — but they are both blank — and [1,4] and [4,5], but `Y V` does not appear on the right-hand side of any rule, so it is also blank.
- For [0,5], there are four possibilities[4]:
  - [0,1] and [1,5], but [1,5] is blank.
  - [0,2] and [2,5], but [2,5] is blank.
  - [0,3] and [3,5], but [3,5] is blank.
  - [0,4] and [4,5]: there are two non-terminals at [0,4], so we must examine both of them. However, neither `NP V` nor `X V` forms the right-hand side of any rule.
- Eventually, all of this column (except [4,5]) is blank.
- Something similar happens for *an*:
  - [5,6] is `Det`.
  - [4,6] is blank, because there is no `V Det`.
  - [3,6] is blank, because [4,6] is blank, as well as [3,5].
  - [2,6] is blank, because [3,6], [4,6], and [2,5] are all blank.
  - [1,6] is blank, because [2,6], [3,6], [4,6], and [1,5] are all blank.
  - [0,6] is blank, because [1,6], [2,6], [3,6], [4,6] and [0,5] are all blank. (See Table 5).
- Next, we have *park* with `N` at [6,7]. [5,7] has `Det N` (from [5,6] and [6,7]), which gives us `NP`.
- [4,7] has `V NP` (from [4,5] and [5,7]), which gives us `VP`, but [4,6] and [6,7] gives us nothing.
- [3,7] has `NP VP` (from [3,4] and [4,7]), which gives us `S`, but [3,5] and [5,7], as well as [3,6] and [6,7] both give us nothing. Note that the `S` here indicates that this fragment "Bob walked an park" is indeed a sentence by itself, but we want to parse the entire sentence "an park by Bob walked an park with Bob", so we need to keep filling in the chart.
- For [2,7], we have:
  - [2,3] and [3,7], but `P S` isn't the right-hand side of any rule.
  - [2,4] and [4,7], but `PP VP` isn't the right-hand side of any rule.

[4] I like to think of this as a sort-of "lever" or "see-saw" procedure, where we read pairs of cells off the table by starting at the left-most ([0,1]) and the neighbour to the bottom ([1,5]) and then proceed right ([0,2]) and down ([2,5]), and so on.

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] ?	[0,5] ?	[0,6] ?	[0,7] ?	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] -	[1,4] ?	[1,5] ?	[1,6] ?	[1,7] ?	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] ?	[2,5] ?	[2,6] ?	[2,7] ?	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] ?	[3,6] ?	[3,7] ?	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] ?	[4,7] ?	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] ?	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 3: Cell [0,3] (spanning "an park by") has no partial parse, because there is no rule for `NP P`

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP X	[0,5] ?	[0,6] ?	[0,7] ?	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] -	[1,4] Y	[1,5] ?	[1,6] ?	[1,7] ?	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] PP	[2,5] ?	[2,6] ?	[2,7] ?	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] ?	[3,6] ?	[3,7] ?	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] ?	[4,7] ?	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] ?	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 4: Cell [0,4] (spanning "an park by Bob") has two competing analyses: `NP->Det Y` (in red) and `X->NP PP` (in blue)

  - [2,5] and [5,7], but [2,5] is blank.
  - [2,6] and [6,7], but [2,6] is blank.
- For [1,7], we have:
  - [1,2] and [2,7], but [2,7] is blank.
  - [1,3] and [3,7], but [1,3] is blank.
  - [1,4] and [4,7], but `Y VP` isn't the right-hand side of any rule.
  - [1,5] and [5,7], but [1,5] is blank.
  - [1,6] and [6,7], but [1,6] is blank.
- For [0,7], we have:
  - [0,1] and [1,7], but [1,7] is blank.
  - [0,2] and [2,7], but [2,7] is blank.
  - [0,3] and [3,7], but [0,3] is blank.
  - [0,4] and [4,7], where there are two possibilities for [0,4], so we consider them both: `NP VP`, which is an `S`, and `X VP`, which isn't in the right-hand side of any rule.
  - [0,5] and [5,7], but [0,5] is blank.
  - [0,6] and [6,7], but [0,6] is blank.
- All in all, we label [0,7] with `S`. There is another fragment here which is itself a sentence: "an park by Bob walked an park". (See Table 6.)
- For the eighth column (under *with*), we have `P` at [7,8]. For [6,8]: we observe that `N P` (from [6,7] and [7,8]) is not productive.
- For [5,8]: [6,8] is blank, and `NP P` (from [5,7] and [7,8]) is not productive.
- For [4,8]: [5,8] and [6,8] are both blank, and `VP P` (from [4,7] and [7,8]) is not productive.
- For [3,8]: [4,8], [5,8], and [6,8] are all blank, and `S P` (from [3,7] and [7,8]) is not productive.
- For [2,8]: all of [3,8], [4,8], [5,8], [6,8], and [2,7] are blank.
- For [1,8]: all of [2,8], [3,8], [4,8], [5,8], [6,8], and [1,7] are blank.
- For [0,8]: all of [1,8], [2,8], [3,8], [4,8], [5,8], and [6,8] are blank, and `S P` (from [0,7] and [7,8]) is not productive.
- For the final column (under *Bob*), we have `NP` at [8,9]. For [7,9]: we have `P NP` (from [7,8] and [8,9]), which is a `PP`.
- For [6,9], we observe that `N PP` is a `Y` (from [6,7] and [7,9]) and [6,8] is blank.
- For [5,9][5], we have:
  - [5,6] and [6,9]: `Det Y` is an `NP`.
  - [5,7] and [7,9]: `NP PP` is an `X`.
  - [5,8] and [8,9]: [5,8] is blank.
- For [4,9], we have:
  - [4,5] and [5,9] represents an interesting case: [5,9] has two possibilities, both of which can combine with the `V` from [4,5] to give a `VP` (`VP->V NP` and `VP->V X`). This is a case of **structural ambiguity**,

[5]You might like to compare this to the very similar cell at [0,4].

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP X	[0,5] -	[0,6] -	[0,7] ?	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] ?	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] ?	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] -	[3,6] -	[3,7] ?	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] -	[4,7] ?	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] ?	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 5: After processing column 6 (“an park by Bob walked an”)

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP X	[0,5] -	[0,6] -	[0,7] S	[0,8] ?	[0,9] ?
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] -	[1,8] ?	[1,9] ?
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] -	[2,8] ?	[2,9] ?
			[3,4] NP	[3,5] -	[3,6] -	[3,7] S	[3,8] ?	[3,9] ?
				[4,5] V	[4,6] -	[4,7] VP	[4,8] ?	[4,9] ?
					[5,6] Det	[5,7] NP	[5,8] ?	[5,9] ?
						[6,7] N	[6,8] ?	[6,9] ?
							[7,8] P	[7,9] ?
								[8,9] NP

Table 6: Cell [0,7] indicates that the fragment “an park by Bob walked an park” is a sentence

where two different trees can be drawn for this VP, as we will see later on.
  - [4,6] and [6,9]: [4,6] is blank.
  - [4,7] and [7,9]: VP Y is not productive.
  - [4,8] and [8,9]: [4,8] is blank.
- For [3,9], we have:
  - [3,4] and [4,9]: the NP at [3,4] can combine with **either** of the (ambiguous) VPs at [4,9] to form an S, so there are, in fact, two S readings here.
  - [3,5] and [5,9]: [3,5] is blank.
  - [3,6] and [6,9]: [3,6] is blank.
  - [3,7] and [7,9]: S Y is not productive.
  - [3,8] and [8,9]: [3,8] is blank.
- For [2,9], we have:
  - [2,3] and [3,9]: (both of the two possible cases of) P S is not productive.
  - [2,4] and [4,9]: (both of the two possible cases of) PP VP is not productive.
  - [2,5] and [5,9]: [2,5] is blank.
  - [2,6] and [6,9]: [2,6] is blank.
  - [2,7] and [7,9]: [2,7] is blank.
  - [2,8] and [8,9]: [2,8] is blank.
- [1,9] will be blank, because all of [2,9], [1,3], [1,5], [1,6], [1,7], amd [1,8] are all blank, and [1,4] and [4,9] give us (two) unproductive Y VP.
- Finally, we have [0,9], which is the span of the entire sentence:
  - [0,1] and [1,9]: [1,9] is blank.
  - [0,2] and [2,9]: [2,9] is blank.
  - [0,3] and [3,9]: [0,3] is blank.
  - [0,4] and [4,9]: each of these has two readings: NP and X for [0,4], and VP (twice) for [4,9]. The latter combination is not productive, but we can make two different S readings based on the former.
  - [0,5] and [5,9]: [0,5] is blank.
  - [0,6] and [6,9]: [0,6] is blank.
  - [0,7] and [7,9]: S PP is not productive.
  - [0,8] and [8,9]: [0,8] is blank.
- Based on the completed chart (see Table 7), we can observe the two parses for this (structurally ambiguous) sentence: both have the same subject (the purple NP at [0,4]), but different predicates depending on whether the prepositional phrase at [7,9] attaches to make a noun phrase at [5,9] (in red, on the left below), or directly attaches to the verb phrase at [4,9] (in blue, on the right below).

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP X	[0,5] -	[0,6] -	[0,7] S -	[0,8] -	[0,9] S S
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] -	[1,8] -	[1,9] -
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] -	[2,8] -	[2,9] -
			[3,4] NP	[3,5] -	[3,6] -	[3,7] S	[3,8] -	[3,9] S, S
				[4,5] V	[4,6] -	[4,7] VP	[4,8] -	[4,9] VP VP
					[5,6] Det	[5,7] NP	[5,8] -	[5,9] NP X
						[6,7] N	[6,8] -	[6,9] Y
							[7,8] P	[7,9] PP
								[8,9] NP

Table 7: Cell [0,9] indicates that there are two parses for the sentence "an park by Bob walked an park with Bob", based on the NP (in purple) and the two different VP readings (in red and blue)

```
(S   (NP   (Det an)
           (N park)
           (PP          (P by)
                        (NP Bob)
           )
     )
     (VP   (V walked)
           (NP          (Det an)
                        (N park)
                        (PP         (P with)
                                    (NP Bob)
                        )
           )
     )
)
```

```
(S   (NP   (Det an)
           (N park)
           (PP          (P by)
                        (NP Bob)
           )
     )
     (VP   (V walked)
           (NP          (Det an)
                        (N park)
           )
           (PP          (P with)
                        (NP Bob)
           )
     )
)
```

Table 8: The two trees for the parses in the table above; left corresponds to the red colour; right corresponds to blue