Parsing
Exercises

Laura Kallmeyer

WS 2016/2017, Heinrich-Heine-Universität Düsseldorf

Question 1 (Grammars)

Consider the following three languages:

- $L_1 = \{a^n b^m c d^m e^n \mid n, m \geq 0\}$
- $L_2 = \{(ab)^n c d^m \mid n, m \geq 0\}$
- $L_3 = \{a^n b (cd)^n e^n \mid n \geq 0\}$

One of the languages is regular, one context-free and not regular and one not context-free. Which are the regular and the non-regular context-free languages? Justify your answer by giving the corresponding grammars.

Solution:

L_2 is regular: $S \to abS, S \to c, S \to cB, B \to d, B \to dB$.

L_1 is context-free: $S \to aSe, S \to T, T \to bTd, T \to c$.

L_3 is context-sensitive:
$S \to GbH$, $S \to b$,
$G \to GA$, $Aa \to aA$, $Ab \to abC$,
$Ccd \to cdC$, $C \to cdE$, $Ee \to eE$, $EH \to eH$
$G \to A'$, $A'a \to aA'$, $A'b \to abC'$,
$C'cd \to cdC'$, $C'e \to cde$, $C'H \to cde$, $H \to e$
(this grammar was not required)

Question 2 (CFG)

1. *Consider the CFG G_1 with non-terminals $\{S, T, A, B\}$, terminals $\{a, b\}$, start symbol S and productions*
   $S \to ATA \quad S \to BTB$
   $T \to ATA \quad T \to BTB \quad T \to \epsilon$
   $A \to a \quad B \to b$

   (a) *Transform G_1 into an equivalent CFG G_1' without ϵ-productions.*

   (b) *Transform G_1' into an equivalent CFG G_1'' in Chomsky Normal Form.*

2. *Consider the CFG G_2 with non-terminals $\{S, A, B\}$, terminals $\{a, b\}$, start symbol S and productions*
   $S \to AB \quad A \to S \quad A \to a \quad B \to b$
   Transform G_2 into an equivalent CFG G_2' without left recursion.

Solution:

1. (a) First, calculate the set N_ϵ of all $A \in N$ such that $A \overset{*}{\Rightarrow} \epsilon$: $N_\epsilon = \{T\}$

   Consequently, the productions in G_1' are

   $S \to ATA \quad S \to BTB \quad S \to AA \quad S \to BB$
   $T \to ATA \quad T \to BTB \quad T \to AA \quad T \to BB$
   $A \to a \quad B \to b$

   (b) For the transformation into CNF, we introduce new non-terminals C_1, C_2. The new set of productions in G_1'' is

   $S \to AC_1 \quad S \to BC_2 \quad S \to AA \quad S \to BB$
   $T \to AC_1 \quad T \to BC_2 \quad T \to AA \quad T \to BB$
   $C_1 \to TA \quad C_2 \to TB \quad A \to a \quad B \to b$

2. We put indices on our non-terminals: B has index 1, A index 2 and S index 3:

   $S_3 \to A_2B_1 \quad A_2 \to S_3 \quad A_2 \to a \quad B_1 \to b$

   Obviously, this grammar is left-recursive: $S_3 \Rightarrow A_2B_1 \Rightarrow S_3B_1$

   For the indices 1 and 2 the condition that every rhs starts either with a terminal or with a non-terminal of higher index is satisfied.

   Consider S_3: in order to remove the problematic production $S_3 \to A_2B_1$, we replace A_2 with the rhs of A_2-productions. Our new productions are

   $S_3 \to S_3B_1 \quad S_3 \to aB_1 \quad A_2 \to S_3 \quad A_2 \to a \quad B_1 \to b$

   Now we have one left-recursive productions, $S_3 \to S_3B_1$, that still needs to be removed:

   We introduce a new non-terminal C and replace $S_3 \to S_3B_1$, $S_3 \to aB_1$

   with $S_3 \to aB_1$, $S_3 \to aB_1C$, $C \to B_1C$, $C \to B_1$.

   As a result, we obtain the following productions:

   $S_3 \to aB_1 \quad S_3 \to aB_1C \quad C \to B_1C \quad C \to B_1 \quad A_2 \to S_3 \quad A_2 \to a \quad B_1 \to b$

   Note that by this transformation, the non-terminal A_2 became useless since it is no longer reachable from the start symbol. Furthermore, we have unary productions.

   If we remove the productions with the useless symbol A_2 and if we eliminate the unary productions, we obtain the productions

   $S_3 \to aB_1 \quad S_3 \to aB_1C \quad C \to B_1C \quad C \to b \quad B_1 \to b$

   We could also start with different indices, e.g.,

   $S_2 \to A_3B_1 \quad A_3 \to S_2 \quad A_3 \to a \quad B_1 \to b$

   Then we would obtain the following productions:

   $S_2 \to A_3B_1 \quad A_3 \to a \quad A_3 \to aC \quad C \to B_1 \quad C \to B_1C \quad B_1 \to b$

   After elimination of the unary production $C \to B_1$, this yields

   $S_2 \to A_3B_1 \quad A_3 \to a \quad A_3 \to aC \quad C \to b \quad C \to B_1C \quad B_1 \to b$

Question 3 (PDA)

Give a PDA that recognizes the following language: $\{a^n b^m c d^m e^n \mid n, m \geq 0\}$.

Solution: PDA $M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ with

- $Q = \{q_1, q_2, q_3, q_4\}$; $\Sigma = \{a, b, c, d, e\}$; $\Gamma = \{\#, E, D\}$;
- q_1 initial state, $\#$ initial stack symbol; $F = \{q_4\}$;
- $\delta(q_1, a, \epsilon) = \{\langle q_1, E\rangle\}$, $\delta(q_1, b, \epsilon) = \{\langle q_2, D\rangle\}$, $\delta(q_1, c, \epsilon) = \{\langle q_3, \epsilon\rangle\}$,
  $\delta(q_2, b, \epsilon) = \{\langle q_2, D\rangle\}$, $\delta(q_2, c, \epsilon) = \{\langle q_3, \epsilon\rangle\}$,
  $\delta(q_3, d, D) = \{\langle q_3, \epsilon\rangle\}$, $\delta(q_3, e, E) = \{\langle q_3, \epsilon\rangle\}$, $\delta(q_3, \epsilon, \#) = \{\langle q_4, \#\rangle\}$.

Question 4 (PDA)

Consider the CFG G with non-terminals $\{S, A, B\}$, terminals $\{a, b\}$, start symbol S and productions

$S \to aSB \quad S \to aB \quad B \to b$

Give the three different PDAs that are equivalent to this grammar and that are described on the PDA slides 12 and 13.

Solution:

1. $M = \langle \{q\}, \{a, b\}, \{S, B\}, \delta, q, S, \emptyset \rangle$ with
   $\delta(q, a, S) = \{\langle q, SB \rangle, \langle q, B \rangle\}$, $\delta(q, b, B) = \{\langle q, \varepsilon \rangle\}$.
   In all other cases, δ yields $\emptyset$.
   Acceptance with the empty stack.
2. $M = \langle \{q_0, q_1, q_f\}, \{a, b\}, \{S, B, a, b, Z_0\}, \delta, q_0, Z_0, \{q_f\} \rangle$ with
   $\delta(q_0, \varepsilon, Z_0) = \{\langle q_1, SZ_0 \rangle\}$,
   $\delta(q_1, \varepsilon, S) = \{\langle q_1, aSB \rangle, \langle q_1, aB \rangle\}$, $\delta(q_1, \varepsilon, B) = \{\langle q_1, b \rangle\}$,
   $\delta(q_1, a, a) = \{\langle q_1, \varepsilon \rangle\}$, $\delta(q_1, b, b) = \{\langle q_1, \varepsilon \rangle\}$,
   $\delta(q_1, \varepsilon, Z_0) = \{\langle q_f, \varepsilon \rangle\}$.
   In all other cases, δ yields $\emptyset$.
   Acceptance in the final state q_f.
3. $M = \langle \{q_0, q_1, q_f\}, \{a, b\}, \{S, B, a, b, Z_0\}, \delta, q_0, Z_0, \{q_f\} \rangle$ with
   $\langle q_0, a \rangle \in \delta(q_0, a, \varepsilon)$, $\langle q_0, b \rangle \in \delta(q_0, b, \varepsilon)$.
   $\langle q_0, S \rangle \in \delta(q_0, \epsilon, BSa)$, $\langle q_0, S \rangle \in \delta(q_0, \epsilon, Ba)$, $\langle q_0, B \rangle \in \delta(q_0, \epsilon, b)$.
   $\langle q_1, \epsilon \rangle \in \delta(q_0, \epsilon, S)$
   $\langle q_f, \epsilon \rangle \in \delta(q_1, \epsilon, Z_0)$
   These are all elements in the values of the δ function.

Question 5 (Unger parser)

1. *Give the pseudocode for the Unger recognizer with tabulation under the assumption that the CFG is in Chomksy normal form.*

   As a notation for substrings of the input $w = w_1 \dots w_n$ ($w_1, \dots, w_n \in T$), use the following pairs of indices: $\langle i, j \rangle$ for $1 \leq i \leq j \leq n$ stands for the substring $w_i \dots w_j$.

   In other words, you have to tabulate results $\langle A, i, j, res \rangle$ whenever a call `unger(`$A, \langle i, j \rangle$`)` *has returned res.*
2. *Extend this pseudocode such that the parser generates a parse forest grammar, i.e., a set of productions of the form $\langle X, \langle i, j \rangle \rangle \to \langle X_1, \langle i_1, j_1 \rangle \rangle \dots \langle X_k, \langle i_k, j_k \rangle \rangle$.*

   For this, we need two global structures that get filled:

   (a) the chart $\mathcal{C}$ that tells us whether a category X with a span $\langle i, j \rangle$ has already been tested and if so, with which result, and

   (b) the list of productions annotated with spans that have been successfully parsed.

Solution:

Since the CFG is in CNF, it does in particular not contain ϵ-productions or unary productions. Consequently, we don't need to check for loops.

Initially, for a given (global) $w = w_1 \dots w_n$, we call the parser with `unger(`$\langle 0, n \rangle, S$`)`

We assume a global set R of already computed results, initialized with $\emptyset$.

1.
```
function unger($\langle i,j \rangle$,$A$):
    $out$ := false;
    if there is a $res$ with $\langle A, i, j, res \rangle \in R$,
    then return $res$;
    else if ($j = i + 1$ and $A \to w_j \in P$),
        then $out$ := true
        else for all $A \to B\,C \in P$:
            for all $k$ with $i < k < j$:
                if unger($\langle i,k \rangle, B$) and unger($\langle k,j \rangle, C$)
                then $out$ := true;
        add $\langle A, i, j, out \rangle$ to $R$;
        return out
```

2. In order to turn this into a parser, we add a set F of span-annotated productions that present the parse forest, initialized with $\emptyset$. The parts that are added are bold:

```
function unger($\langle i,j \rangle$,$A$):
    $out$ := false;
    if there is a $res$ with $\langle A, i, j, res \rangle \in R$,
    then return $res$;
    else if ($j = i + 1$ and $A \to w_j \in P$),
        then add $\langle A, \langle i,j \rangle \rangle \to \langle w_j, \langle i,j \rangle \rangle$ to $F$;
            $out$ := true
        else for all $A \to B\,C \in P$:
            for all $k$ with $i < k < j$:
                if unger($\langle i,k \rangle, B$) and unger($\langle k,j \rangle, C$)
                then add $\langle A, \langle i,j \rangle \rangle \to \langle B, \langle i,k \rangle \rangle \, \langle C, \langle k,j \rangle \rangle$ to $F$;
                    $out$ := true;
        add $\langle A, i, j, out \rangle$ to $R$;
        return out
```

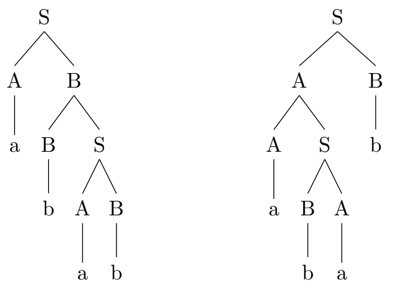
Question 6 (Top-Down Parsing)

Consider a CFG with non-terminals $\{S, A, B\}$, terminals $\{a, b\}$, start symbol S and the following productions: $S \to AB \mid BA, B \to b \mid BS, A \to a \mid AS$.

1. *Give the parse trees for $w = abab$.*
2. *Give the sequence of triples of remaining input, analysis and prediction stack that arises when performing a directional top-down parsing with this grammar with a depth-first strategy.*

Solution:

1.



2.

input	analysis stack	stack
abab		S
abab	S_1	AB
abab	S_1A_1	aB
bab	S_1A_1a	B
bab	$S_1A_1aB_1$	b
ab	$S_1A_1aB_1b$	ε
bab	$S_1A_1aB_2$	BS
bab	$S_1A_1aB_2B_1$	bS
ab	$S_1A_1aB_2B_1b$	S
ab	$S_1A_1aB_2B_1bS_1$	AB
ab	$S_1A_1aB_2B_1bS_1A_1$	aB
b	$S_1A_1aB_2B_1bS_1A_1a$	B
b	$S_1A_1aB_2B_1bS_1A_1aB_1$	b
ε	$S_1A_1aB_2B_1bS_1A_1aB_1b$	ε

Question 7 (Top-down Parsing with deduction rules)

Consider a CFG with the following productions: $S \to aB \mid bA, A \to a \mid aS \mid bAA, B \to b \mid bS \mid aBB$.

Consider the input $w = abba$ *and the deduction rules for top-down parsing.*

1. *Give all items the parser generates for this input. For every item, indicate the rule that was used to deduce this item and indicate the antecedent items of this rule.*
2. *How does the parser know whether* $w = abba$ *is in the language generated by the grammar?*

Solution:

1.

id	item	operation	antecedent items
1	$[S, 0]$	axiom	–
2	$[aB, 0]$	predict	1
3	$[bA, 0]$	predict	1
4	$[B, 1]$	scan	2
5	$[b, 1]$	predict	4
6	$[bS, 1]$	predict	4
7	$[aBB, 1]$	predict	4
8	$[\varepsilon, 2]$	scan	5
9	$[S, 2]$	scan	6
10	$[aB, 2]$	predict	9
11	$[bA, 2]$	predict	9
12	$[A, 3]$	scan	10
13	$[a, 3]$	predict	12
14	$[aS, 3]$	predict	12
15	$[bAA, 3]$	predict	12
16	$[\varepsilon, 4]$	scan	13
17	$[S, 4]$	scan	14
18	$[aB, 4]$	predict	17
19	$[bA, 4]$	predict	17

2. There is a goal item $[\varepsilon, 4]$ in the chart, therefore the word is in the language.

Question 8 (Unger with deduction rules)

Consider a CFG with the following productions: $S \to aSc \mid aT \mid ac, T \to cT \mid c$.

Consider the input $w = ac$ *and the deduction rules for non-directional top-down parsing (= Unger parsing).*

1. *Give all items the parser generates for this input. For every item, indicate the rule that was used to deduce this item and indicate the antecedent items of this rule.*
2. *How does the parser know whether $w = ac$ is in the language generated by the grammar?*

Solution:

1.

id	item	operation	antecedent items
1	$[\bullet S, 0, 2]$	axiom	–
2	$[\bullet a, 0, 1]$	predict	1
3	$[\bullet T, 1, 2]$	predict	1
4	$[\bullet c, 1, 2]$	predict	1
5	$[a\bullet, 0, 1]$	scan	2
6	$[c\bullet, 1, 2]$	scan	4
7	$[T\bullet, 1, 2]$	complete	3,6
8	$[S\bullet, 0, 2]$	complete	1,5,6 or 1,5,7

2. There is a goal item $[S\bullet, 0, 2]$ in the chart, therefore the word is in the language.

Question 9 (Unger deduction rules for CNF)

Consider the Unger Parser for CFGs in Chomsky Normal Form. Define

$First(A) = \{a \mid a \in T, A \overset{*}{\Rightarrow} a\alpha \text{ for some } \alpha \in (N \cup T)^*\}$

$Last(A) = \{a \mid a \in T, A \overset{*}{\Rightarrow} \alpha a \text{ for some } \alpha \in (N \cup T)^*\}$

Assume that for a given CFG in CNF, for all non-terminals A, the sets $First(A)$ and $Last(A)$ are precompiled and can be used to restrict the Unger predictions.

Give the deduction rules for the Unger Parser for CFGs in CNF where the predictions are constrained by the sets First and Last.

Solution:

Predict: $\dfrac{[\bullet A, i, k]}{[\bullet B, i, j], [\bullet C, j, k]} \quad A \to BC \in P, i < j < k, w_j \in Last(B), w_{j+1} \in First(C)$

Scan: $\dfrac{[\bullet A, i, i+1]}{[A\bullet, i, i+1]} \quad A \to w_{i+1} \in P$

Complete: $\dfrac{[\bullet A, i, k], [B\bullet, i, j], [C\bullet, j, k]}{[A\bullet, i, k]} \quad A \to BC \in P$

Question 10 (CYK recognition – general version)

Consider the CFG with non-terminals S, A, C, terminals a, b, start symbol S and productions $S \to ASC$, $S \to \epsilon$, $A \to a$, $A \to b$, $C \to c$.

Give the chart (the $(n + 1) \times (n + 1)$-table) that results from the general CYK algorithm for the input abaccc.

Solution:

6	S						
5							
4		S					
3							
2			S				
1	a, A	b, A	a, A	c, C	c, C	c, C	
0	S	S	S	S	S	S	S
	1	2	3	4	5	6	7

Question 11 (CYK parsing for CNF grammars)

Consider the CFG with non-terminals S, T, A, B, C, D, terminals a, b, start symbol S and productions $S \to AB$, $S \to CT$, $T \to SD$, $A \to AA$, $A \to a$, $B \to BB$, $B \to b$, $C \to a$, $D \to b$.

This grammar is in Chomsky Normal Form.

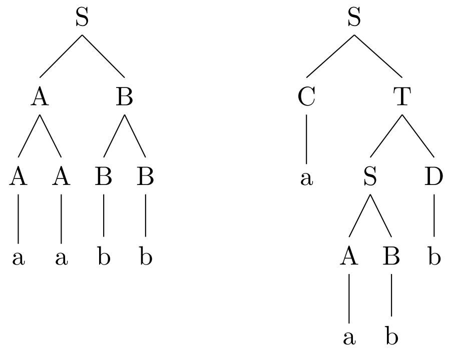
1. *Give the chart (the $n \times n$-table) that results from the CYK parsing algorithm (for CNF) for the input aabb. The chart should include not only the non-terminals that we find but the entire productions with, in the rhs, the indices of the antecedent chart items in the complete rule that has been applied.*
2. *Give all parse trees for the input.*

Solution:

1. Chart:

4	$S \to A_{1,2}B_{3,2}$, $S \to C_{1,1}T_{2,3}$, $T \to S_{1,3}D_{4,1}$			
3	$S \to A_{1,2}B_{3,1}$	$S \to A_{2,1}B_{3,2}$, $T \to S_{2,2}D_{4,1}$		
2	$A \to A_{1,1}A_{2,1}$	$S \to A_{2,1}B_{3,1}$	$B \to B_{3,1}B_{4,1}$	
1	$A \to a$, $C \to a$	$A \to a$, $C \to a$	$B \to b$, $D \to b$	$B \to b$, $D \to b$
	1 a	2 a	3 b	4 b

2. parse trees:



Question 12 (Shift-reduce)

Consider a CFG with the start symbol VP and the following productions:

VP → V NP, VP → VP PP, V → sees,

NP → Det N, Det → the, N → N PP, N → girl, N → telescope,

PP → P NP, P → with

Give all items (pairs of stack and index) that one obtains when doing a directional bottom-up parsing (shift-reduce parsing) of the input the girl with the telescope.

We assume that whenever a terminal is shifted, we perform a reduce in the next step. (This is due to the fact that terminal symbols appear in this grammar only in right-hand sides of length 1.)

Is the input in the language generated by the CFG?

Solution:

	stack	index	operation
1.	ε	0	
2.	the	1	shift
3.	Det	1	reduce 2.
4.	Det girl	2	shift
5.	Det N	2	reduce 4.
6.	NP	2	reduce 5.
7.	Det N with	3	shift 5.
8.	NP with	3	shift 6.
9.	Det N P	3	reduce 7.
10.	NP P	3	reduce 8.
continue with 9:			
11.	Det N P the	4	shift 9.
12.	Det N P Det	4	reduce 11.
13.	Det N P Det telescope	5	shift 12.
14.	Det N P Det N	5	reduce 13.
15.	Det N P NP	5	reduce 14.
16.	Det N PP	5	reduce 15.
17.	Det N	5	reduce 16.
18.	NP	5	reduce 17.
continue with 10:			
	...		
19.	NP PP	5	

No goal item (stack VP) obtained, therefore the input is not in the language.

Question 13 (LL(1) grammar)

Consider a CFG with the following productions: $S \to AB, A \to aAa, A \to \epsilon, B \to bBb, B \to \epsilon$.

Is this grammar LL(1)?

Solution: We need to check whether for all $A \in N$ with $A \to \alpha_1 | \dots | \alpha_n$ being all A-productions in G, the following holds: a) $First(\alpha_1)$, ..., $First(\alpha_n)$ are pairwise disjoint, and b) if $\epsilon \in First(\alpha_j)$ for some $j \in [1..n]$, then $Follow(A) \cap First(\alpha_i) = \emptyset$ for all $1 \leq i \leq n, j \neq i$ (see slide 6).

The $First$ and $Follow$ sets of the non-terminals are

$First(A) = \{\epsilon, a\}$, $First(B) = \{\epsilon, b\}$, $First(S) = \{\epsilon, a, b\}$.

The $Follow$ sets of the non-terminals are as follows:

$Follow(S) = \{\$\}$, $Follow(A) = \{a, b, \$\}$, $Follow(B) = \{b, \$\}$.

Check of the conditions:

- For S, the condition is trivially fulfilled since there is only one S-production.
- For A, $First(aAa) = \{a\}$ and $First(\epsilon) = \{\epsilon\}$ are disjoint.

  But: $First(aAa) = \{a\}$ and $Follow(A) = \{a, b, \$\}$ are not disjoint: $\{a\} \cap \{a, b, \$\} = \{a\}$. Therefore the grammar is not LL(1).
- For B, similarly, $First(bBb) = \{b\}$ and $First(\epsilon) = \{\epsilon\}$ are disjoint.

  But: $First(bBb) = \{b\}$ and $Follow(B) = \{b, \$\}$ are not disjoint: $\{b\} \cap \{b, \$\} = \{b\}$.

Question 14 (Left Corner)

Consider a CFG with the following productions: $S \to A \,|\, BU, A \to aA \,|\, a, B \to bB \,|\, b, U \to aUa \,|\, aa$.

Given an input word aa, give the Left Corner Recognition trace, i.e, the set of stack triples, for this input. We assume a <u>*Reduce*</u> *operation with lookahead, i.e.,* <u>*Reduce*</u> *with a new X-production is applied only if the topmost symbol Y of the stack of predicted categories stands in the relation LC^* to X, i.e., $Y \overset{*}{\Rightarrow} X \dots$.*

Solution:

	Γ_{compl}	Γ_{td}	Γ_{lhs}	operation	
1.	aa	S	–		
2.	a	\$S	A	reduce from 1., $A \to a$	
3.	a	A\$S	A	reduce from 1., $A \to aA$	
4.	Aa	S	–	move from 2.	
5.		\$A\$S	AA	reduce from 3., $A \to a$	
6.		A\$A\$S	AA	reduce from 3., $A \to aA$	failure
7.	a	\$S	S	reduce from 4., $S \to A$	
8.	Sa	S	–	move from 7.	
9.	a	–	–	remove from 8.	failure
10.	A	A\$S	A	move from 5.	
11.		\$S	A	remove from 10.	
12.	A	S		move from 11.	
13.		\$S	S	reduce from 12., $S \to A$	
14.	S	S	–	move from 13.	
15.	–	–	–	remove from 14.	success

Question 15 (Left Corner chart parsing)

Consider the left corner chart parsing deduction rules from slide 14. Extend the algorithm with a rule for ε-productions in order to make it work for arbitrary CFGs.

Solution:

We need the following additional rule:

ε-Scan: $$\frac{}{[A, i, 0]} \quad A \to \varepsilon \in P, 1 \leq i \leq n+1$$

Question 16 (Earley Parsing/recognition)

Consider the CFG $G_3 = \langle N, T, P, S\rangle$ with $N = \{S, A, B, X\}$, $T = \{a, b\}$, $P = \{S \to ABA,\ S \to aXa,\ X \to bXb,\ X \to \epsilon,\ A \to a,\ A \to aA,\ B \to bb\}$

Give the chart resulting from an Earley-recognition of abba with prediction lookahead and completion lookahead:

Predict with lookahead: $$\frac{[A \to \alpha \bullet B\beta, i, j]}{[B \to \bullet\gamma, j, j]} \quad B \to \gamma \in P, w_{i+1} \in First(\gamma) \text{ or } \epsilon \in First(\gamma)$$

Complete with lookahead: $$\frac{[A \to \alpha \bullet B\beta, i, j], [B \to \gamma\bullet, j, k]}{[A \to \alpha B \bullet \beta, i, k]} \quad w_{k+1} \in First(\beta) \text{ or } \epsilon \in First(\beta)$$

Solution:

4	$S \to ABA\bullet$ $S \to aXa\bullet$			$A \to a \bullet A$ $A \to a\bullet$	
3	$S \to aX \bullet a$ $S \to AB \bullet A$	$B \to bb\bullet$ $X \to bXb\bullet$	$X \to b \bullet Xb$	$A \to \bullet aA$ $A \to \bullet a$ $X \to \bullet$	
2		$X \to bX \bullet b$ $X \to b \bullet Xb$ $B \to b \bullet b$	$X \to \bullet$ $X \to \bullet bXb$		
1	$S \to A \bullet BA$ $A \to a\bullet$ $A \to a \bullet A$ $S \to a \bullet Xa$	$B \to \bullet bb$ $X \to \bullet$ $X \to \bullet bXb$			
0	$A \to \bullet a$ $A \to \bullet aA$ $S \to \bullet aXa$ $S \to \bullet ABA$				
	0	1	2	3	4

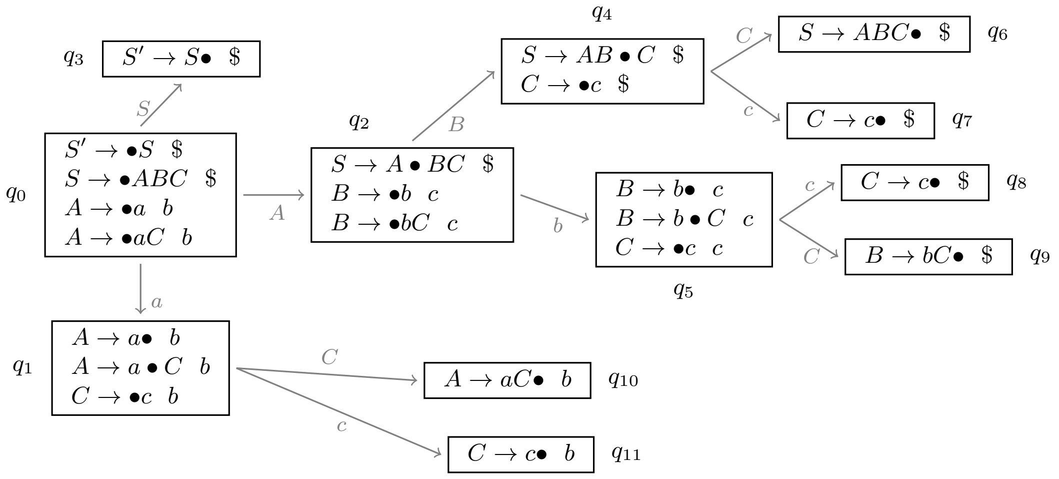
Question 17 (LR parsing)

Consider the CFG $G_4 = \langle N, T, P, S\rangle$ with $N = \{S, A, B, C\}$, $T = \{a, b, c\}$ and productions $1.S \to ABC$, $2.A \to a$, $3.A \to aC$, $4.B \to b$, $5.B \to bC$, $6.C \to c$. This grammar is not LR(1).

1. *Construct the LR(1) states and transitions with the canonical LR algorithm.*
2. *From this, construct the LR(1) parse table with multiple entries for some of the fields.*

Solution:

1.



2. Parse table:

	a	b	c	\$	A	B	C	S
0	s1				2			3
1		r2	s11				10	
2		s5				4		
3				acc				
4			s7				6	
5			s8, r4				9	
6				r1				
7				r6				
8			r6					
9			r5					
10		r3						
11		r6						

Question 18 (Tomita)

The following table is the LR(1) parse table for the CFG with non-terminals $\{A, B, X\}$*, terminals* $\{a, b\}$*, start symbol* S *and productions 1.* $S \rightarrow ABA$*, 2.* $S \rightarrow aXa$*, 3.* $X \rightarrow bXb$*, 4.* $X \rightarrow \epsilon$*, 5.* $A \rightarrow a$*, 6.* $A \rightarrow aA$*, 7.* $B \rightarrow bb$

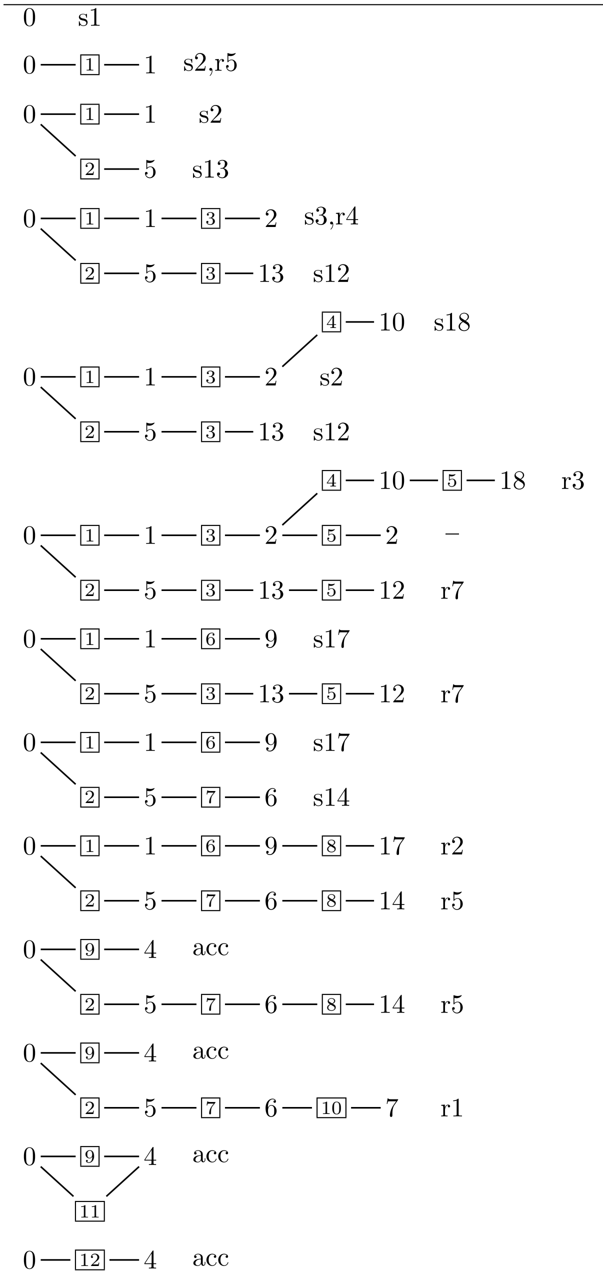
(The table has multiple entries for some of the fields.)

	a	*b*	*\$*	*S*	*A*	*B*	*X*
0	*s1*			*4*	*5*		
1	*s8,r4*	*s2,r5*			*16*		*9*
2		*s3,r4*					*10*
3		*s3,r4*					*11*
4			*acc*				
5		*s13*				*6*	
6	*s14*				*7*		
7			*r1*				
8	*s8*	*r5*			*16*		
9	*s17*						
10		*s18*					
11		*s19*					
12	*r7*						
13		*s12*					
14	*s14*		*r5*		*15*		
15			*r6*				
16		*r6*					
17			*r2*				
18	*r3*						
19		*r3*					

Give the trace of the Tomita-parse for abba (with all intermediate stack graphs and all analyses).

Solution:

Stack analysis


0 s1
0 — $\boxed{1}$ — 1 s2,r5
0 — $\boxed{1}$ — 1 s2
$\boxed{2}$ — 5 s13
0 — $\boxed{1}$ — 1 — $\boxed{3}$ — 2 s3,r4
$\boxed{2}$ — 5 — $\boxed{3}$ — 13 s12
$\boxed{4}$ — 10 s18
0 — $\boxed{1}$ — 1 — $\boxed{3}$ — 2 s2
$\boxed{2}$ — 5 — $\boxed{3}$ — 13 s12
$\boxed{4}$ — 10 — $\boxed{5}$ — 18 r3
0 — $\boxed{1}$ — 1 — $\boxed{3}$ — 2 — $\boxed{5}$ — 2 –
$\boxed{2}$ — 5 — $\boxed{3}$ — 13 — $\boxed{5}$ — 12 r7
0 — $\boxed{1}$ — 1 — $\boxed{6}$ — 9 s17
$\boxed{2}$ — 5 — $\boxed{3}$ — 13 — $\boxed{5}$ — 12 r7
0 — $\boxed{1}$ — 1 — $\boxed{6}$ — 9 s17
$\boxed{2}$ — 5 — $\boxed{7}$ — 6 s14
0 — $\boxed{1}$ — 1 — $\boxed{6}$ — 9 — $\boxed{8}$ — 17 r2
$\boxed{2}$ — 5 — $\boxed{7}$ — 6 — $\boxed{8}$ — 14 r5
0 — $\boxed{9}$ — 4 acc
$\boxed{2}$ — 5 — $\boxed{7}$ — 6 — $\boxed{8}$ — 14 r5
0 — $\boxed{9}$ — 4 acc
$\boxed{2}$ — 5 — $\boxed{7}$ — 6 — $\boxed{10}$ — 7 r1
0 — $\boxed{9}$ — 4 acc
$\boxed{11}$
0 — $\boxed{12}$ — 4 acc

$\boxed{1}$: a

$\boxed{2}$: A($\boxed{1}$)

$\boxed{3}$: b

$\boxed{4}$: X(ε)

$\boxed{5}$: b

$\boxed{6}$: X($\boxed{3}$,$\boxed{4}$,$\boxed{5}$)

$\boxed{7}$: B($\boxed{3}$,$\boxed{5}$)

$\boxed{8}$: a

$\boxed{9}$: S($\boxed{1}$,$\boxed{6}$,$\boxed{8}$)

$\boxed{10}$: A($\boxed{8}$)

$\boxed{11}$: S($\boxed{2}$,$\boxed{7}$,$\boxed{10}$)

$\boxed{12}$: [$\boxed{11}$,$\boxed{9}$]

Question 19 (PCFG)

Consider the PCFG G with non-terminals $\{S, A, B\}$, terminals $\{a, b\}$, start symbol S and productions

$$\{ \quad \begin{array}{ll} 0{,}5 & S \to AS, \\ 0{,}3 & S \to SB, \\ 0{,}2 & S \to AB, \\ 1 & A \to a, \\ 1 & B \to b \end{array} \quad \}$$

(The numbers preceding the productions are the corresponding probabilities.)

1. *Give the inside chart for the input $w = aaabbb$.*

2. *Give the viterbi chart of a probabilistic CYK parsing of $w = aaabbb$.*

Solution:

1.

6	$(S, 0.027)$	$(S, 0.027)$	$(S, 0.018)$			$(B, 1)$
5	$(S, 0.045)$	$(S, 0.06)$	$(S, 0.06)$		$(B, 1)$	
4	$(S, 0.005)$	$(S, 0.1)$	$(S, 0.2)$	$(B, 1)$		
3			$(A, 1)$			
2		$(A, 1)$				
1	$(A, 1)$					
	1	2	3	4	5	6

2.

6	$0.0045 : S \rightarrow AS, 1$					
5	$0.015 : S \rightarrow AS, 1$	$0.009 : S \rightarrow AS, 1$				
4	$0.05 : S \rightarrow AS, 1$	$0.03 : S \rightarrow AS, 1$	$0.018 : S \rightarrow SB, 3$			
3		$0.1 : S \rightarrow AS, 1$	$0.06 : S \rightarrow SB, 2$			
2			$0.2 : S \rightarrow AB, 1$			
1	$1 : A \rightarrow a$	$1 : A \rightarrow a$	$1 : A \rightarrow a$	$1 : B \rightarrow b$	$1 : B \rightarrow b$	$1 : B \rightarrow b$
	1	2	3	4	5	6

(For some fields of this chart, there are actually several possibilities leading to the same probability.)

Question 20 (A* parsing)

Consider the PCFG given in the example on slides 14 (A* slides) and the outside scores computed on the subsequent slides.

As input consider "red ugly camping car".

1. *Show the weighted deductive CYK-Parsing with chart and agenda using this grammar and input with weights as described on slide 18 (incorporating the viterbi inside score and the SX outside estimate).*

   Write each weight as a pair (in, out) where in is the inside viterbi score and out the outside estimate (using $|log(p)|$ instead of p).

   Concerning the chart column, it is enough to list only new items in each row. (This is different from the agenda where items are not only added but also removed and reordering depending on weights takes place.)

2. *The log used here is log_{10}. Compute the probability of the best parse tree from the weight of the goal item.*

Solution:

1.

Chart	Agenda
	(0.6,3.8):[A, 1, 2], (0.7,3.8):[A, 0, 1], (0.7,4.1):[N, 2, 3], (1,3.8):[N, 0, 1], (1,4.1):[N, 3, 4]
(0.6,3.8):[A, 1, 2]	(0.7,3.8):[A, 0, 1], (0.7,4.1):[N, 2, 3], (1,3.8):[N, 0, 1], (1,4.1):[N, 3, 4]
(0.7,3.8):[A, 0, 1]	(0.7,4.1):[N, 2, 3], (1,3.8):[N, 0, 1], (1,4.1):[N, 3, 4]
(0.7,4.1):[N, 2, 3]	(1,3.8):[N, 0, 1], (0.6+0.7+0.7,2.9):[N, 1, 3], (1,4.1):[N, 3, 4]
(1,3.8):[N, 0, 1]	(2,2.9):[N, 1, 3], (1,4.1):[N, 3, 4]
(2,2.9):[N, 1, 3]	(1,4.1):[N, 3, 4], ($min\{0.7 + 2 + 0.7, 1 + 2 + 1\}$,1.7):[N, 0, 3]
(1,4.1):[N, 3, 4]	(3.4,1.7):[N, 0, 3], (2+1+1,1.2):[N, 1, 4], (0.7+1+1,2.9):[N, 2, 4]
(3.4,1.7):[N, 0, 3]	(4,1.2):[N, 1, 4], (3.4+1+1,0):[N, 0, 4], (2.7,2.9):[N, 2, 4]
(4,1.2):[N, 1, 4]	(5.4,0):[N, 0, 4], (2.7,2.9):[N, 2, 4]

The last operation does not add to the agenda because all the new items one could possibly build (combining [N, 1, 4] with [A, 0, 1] or [N, 0, 1]) already exist in the agenda and the weights of the new items are higher or equal to the one of the already existing.

Algorithm stops because goal item [N, 0, 4] has been reached as top agenda item.

2. The inside score in the weight of the goal item [N, 0, 4] is 5.4. The probability of the best parse tree is therefore $10^{-5.4} = \frac{1}{10^{5.4}} = 3.98 \cdot 10^{-6} \approx 4 \cdot 10^{-6}$.

Question 21 (A* parsing) *Consider the PCFG $G = \langle N, T, P, S, p\rangle$ with $N = \{S, A, B\}$, $T = \{a, b\}$ and*

$$\begin{array}{rcl} P = \{0,3 & S \rightarrow & AB \\ 0,7 & S \rightarrow & BA \\ 0,1 & A \rightarrow & AS \\ 0,9 & A \rightarrow & a \\ 0,6 & B \rightarrow & BS \\ 0,4 & B \rightarrow & b\}. \end{array}$$

(The numbers preceding the rules are the corresponding probabilities.)

Compute the estimates of the inside viterbi scores $in(X, l)$ for non-terminals $X \in N$ and lengths $1 \le l \le 4$.

Use the following values for the weights:

$$\begin{array}{lll} |\log(0,1)| = 1,00 & |\log(0,3)| = 0,52 & |\log(0,4)| = 0,40 \\ |\log(0,6)| = 0,22 & |\log(0,7)| = 0,15 & |\log(0,9)| = 0,05 \end{array}$$

Solution:

S	∞	$0,6$	∞	$1,42$	
A	$0,05$	∞	$1,65$	∞	
B	$0,40$	∞	$1,22$	∞	
	1	2	3	4	l

Question 22 (A* parsing)

Consider the PCFG G with $N = \{S, A\}$, $T = \{a\}$, start symbol S and productions

0.5	$S \rightarrow SS$	*0.125*	$S \rightarrow AS$	*0.25*	$S \rightarrow SA$
0.125	$S \rightarrow a$	*1*	$A \rightarrow a$		

For weights, use $|log_2(p)|$.

1. *Compute the inside viterbi estimates for lengths $1 \le l \le 4$ and the outside SX estimates for length $n = 4$.*
2. *Use these values for an A* parsing of aaaa.*

Solution:

1. Inside estimates:

S	3	5	7	9	
A	0	∞	∞	∞	
	1	2	3	4	l

Outside SX estimates:

- $l = 4$:
$out(A, 0, 4, 0) = \infty$, $out(N, 0, 4, 0) = 0$

- $l = 3$:
  $out(A, 0, 3, 1) = 3 + 3 = 6$
  $out(A, 1, 3, 0) = 3 + 2 = 5$
  $out(S, 0, 3, 1) = min\{4, 2\} = 2$
  $out(S, 1, 3, 0) = min\{4, 3\} = 3$
- $l = 2$:
  $out(A, 0, 2, 2) = min\{3 + 3 + 2, 3 + 5 + 0\} = 8$
  $out(A, 1, 2, 1) = min\{2 + 3 + 2, 3 + 3 + 3\} = 7$
  $out(A, 2, 2, 0) = min\{2 + 3 + 3, 2 + 5 + 0\} = 7$
  $out(S, 0, 2, 2) = min\{1 + 3 + 2, 1 + 5 + 0, 2 + 0 + 2\} = 4$
  $out(S, 1, 2, 1) = min\{2 + 0 + 3, 3 + 0 + 2, 1 + 3 + 2, 1 + 3 + 3\} = 5$
  $out(S, 2, 2, 0) = min\{1 + 3 + 3, 1 + 5 + 0, 3 + 0 + 3\} = 6$
- $l = 1$:
  $out(A, 0, 1, 3) = min\{3 + 3 + 4, 3 + 5 + 2, 3 + 7 + 0\} = 10$
  $out(A, 1, 1, 2) = min\{3 + 3 + 5, 3 + 5 + 3, 2 + 3 + 4\} = 9$
  $out(A, 2, 1, 1) = min\{3 + 3 + 4, 2 + 3 + 5, 2 + 5 + 2\} = 9$
  $out(A, 3, 1, 0) = min\{2 + 3 + 6, 2 + 5 + 3, 2 + 7 + 0\} = 9$
  $out(S, 0, 1, 3) = min\{1 + 3 + 4, 1 + 5 + 2, 1 + 7 + 0, 2 + 0 + 4\} = 6$
  $out(S, 1, 1, 2) = min\{1 + 3 + 5, 1 + 5 + 3, 1 + 3 + 4, 2 + 0 + 5, 3 + 0 + 4\} = 7$
  $out(S, 2, 1, 1) = min\{1 + 3 + 5, 1 + 3 + 6, 1 + 5 + 2, 2 + 0 + 6, 3 + 0 + 5\} = 8$
  $out(S, 3, 1, 0) = min\{1 + 3 + 6, 1 + 5 + 3, 1 + 7 + 0, 3 + 0 + 6\} = 8$

2. Parsing of *aaaa*:

Chart	Agenda
	(0,9):[A, 1, 2], (0,9):[A, 2, 3], (0,9):[A, 3, 4], (3,6):[S, 0, 1], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4]
(0,9):[A, 1, 2]	(0,9):[A, 2, 3], (0,9):[A, 3, 4], (3,6):[S, 0, 1], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4]
(0,9):[A, 2, 3]	(0,9):[A, 3, 4], (3,6):[S, 0, 1], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4]
(0,9):[A, 3, 4]	(3,6):[S, 0, 1], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4]
(3,6):[S, 0, 1]	(3+0+2,4):[S, 0, 2], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4]
(5,4):[S, 0, 2]	(5+0+2,2):[S, 0, 3], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4],
(7,2):[S, 0, 3]	(7+0+2,0):[S, 0, 4], (0,10):[A, 0, 1], (3,7):[S, 1, 2], (3,8):[S, 2, 3], (3,8):[S, 3, 4],

Parser stops since top agenda item is a goal item.