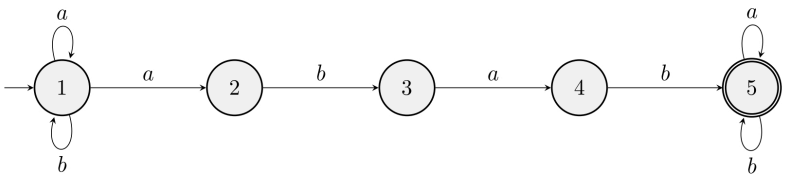
Lecture 4

Exercise Sheet 4

Exercise 1 *Give a context free grammar for the set of palindromes over the alphabet* {`a`, `b`}.

Exercise 2 *Let's consider an NFA that accepts any string that contains the substring "abab".*



1. *Convert the above NFA into its equivalent total DFA.*
2. *Convert the resultant DFA in an equivalent CFG. It is suggested to minimize the DFA before writing CFG.*

Exercise 3 *Consider the language generated by the grammar*

$$\Rightarrow S \quad ::= \quad \texttt{b}SS \quad | \quad \texttt{a}S \quad | \quad \texttt{a}$$

and the string `aabbaaa`.

1. *Find a leftmost derivation for this string.*
2. *Draw the derivation tree.*

Exercise 4 *For the grammar given in Exercise 3, prove that every accepted word has more* `a`*'s than* `b`*'s using structural induction.*

Exercise 5 *Let's look at a "Natural Language" example. The alphabet is*

{ *the, a, cat, dog, happy, tired, slept, died, ate, dinner, and, .* }

The grammar is

Sentence	$\Rightarrow S$	::=	*C .*
Clause	*C*	::=	*NP VP* | *C and C*
Noun phrase	*NP*	::=	*Art N* | *dinner*
Noun	*N*	::=	*Adj N* | *cat* | *dog*
Adjective	*Adj*	::=	*happy* | *tired*
Verb phrase	*VP*	::=	*VI* | *VT NP*
Intransitive verb	*VI*	::=	*slept* | *died*
Transitive verb	*VT*	::=	*ate*
Article	*Art*	::=	*a* | *the*

This grammar accepts "words" such as

the happy tired happy dog died and the cat slept.
the tired tired cat ate dinner.
dinner ate a happy dog.

Try writing derivations and derivation trees for these sentences.

Exercise 6 *Show that the following grammar is ambiguous. The alphabet is* $\{\mathtt{a}, \mathtt{b}\}$.

$$\begin{array}{rrl} \Rightarrow P & ::= & \varepsilon \mid Q\mathtt{a} \mid \mathtt{a}Q \\ Q & ::= & \mathtt{aa}P \mid \mathtt{b}R \\ R & ::= & Q\mathtt{a} \end{array}$$

Exercise 7 *Convert the following CFG into an equivalent CFG in Chomsky normal form*

$$\begin{array}{rrl} \Rightarrow A & ::= & BAB \mid B \mid \varepsilon \\ B & ::= & 00 \mid \varepsilon \end{array}$$

Exercise 8 *Give grammars for the following two languages:*

1. *All binary strings with both an even number of zeroes and an even number of ones.*
2. *All strings of the form* $0^a1^b0^c$ *where* $a + c = b$.

Exercise Sheet 4

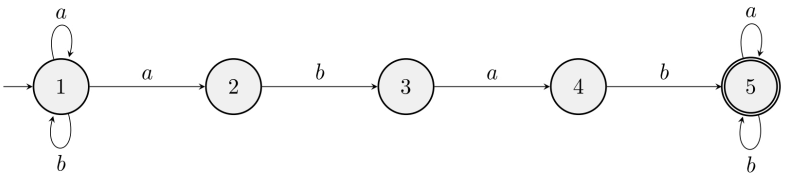
Exercise 1 *Give a context free grammar for the set of palindromes over the alphabet* $\{a, b\}$.

Solution

A CFG for the set of palindromes over the alphabet $\{\texttt{a}, \texttt{b}\}$ is given below:

$$\Rightarrow S \quad ::= \quad \varepsilon \mid \texttt{a} \mid \texttt{b} \mid \texttt{aSa} \mid \texttt{bSb}$$

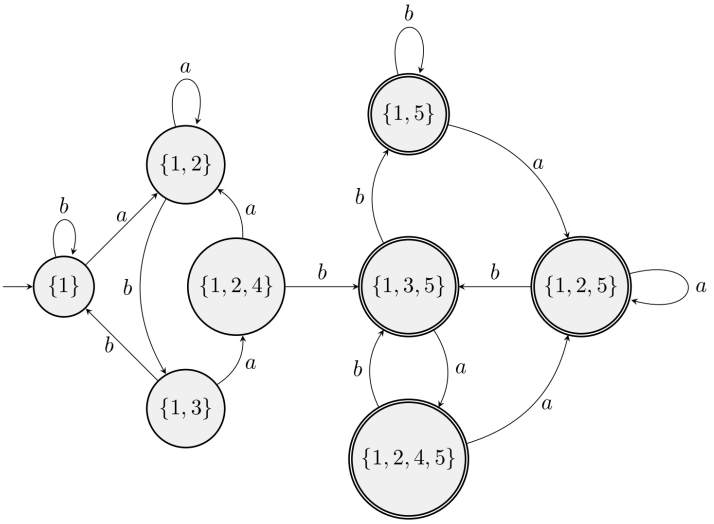
Exercise 2 *Let's consider an NFA that accepts any string that contains the substring "abab".*



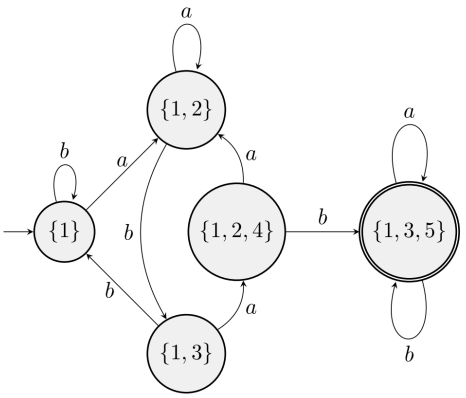
1. *Convert the above NFA into its equivalent total DFA.*
2. *Convert the resultant DFA in an equivalent CFG. It is suggested to minimize the DFA before writing CFG.*

Solution

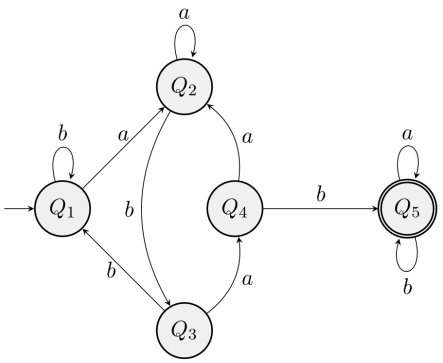
1. On converting the NFA to its equivalent total DFA, we get the following:



2. We can see that the states $\{1, 3, 5\}$, $\{1, 5\}$, $\{1, 2, 4, 5\}$, and $\{1, 2, 5\}$ are equivalent and could be removed. On minimizing, we get the following DFA:



We can rename the states, to make it easier for us to write the equivalent CFG.



Now, we can very easily write the CFG for the above DFA:

$$\begin{array}{rcl} R_1 & ::= & aR_2 \mid bR_1 \\ R_2 & ::= & aR_2 \mid bR_3 \\ R_3 & ::= & aR_4 \mid bR_1 \\ R_4 & ::= & aR_2 \mid bR_5 \\ R_5 & ::= & aR_5 \mid bR_5 \mid \varepsilon \\ \text{Start:} & & R_1 \end{array}$$

Exercise 3 *Consider the language generated by the grammar*

$$\Rightarrow S \quad ::= \quad \texttt{b}SS \quad \mid \quad \texttt{a}S \quad \mid \quad \texttt{a}$$

and the string `aabbaaa`*.*

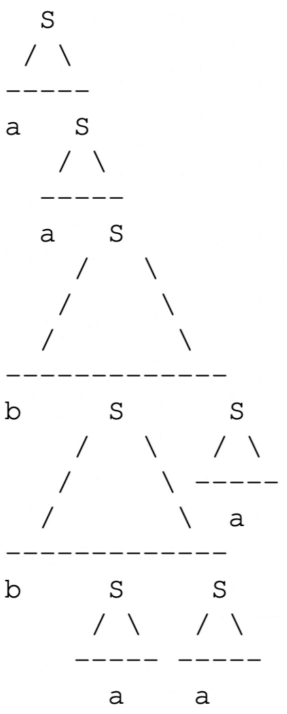
1. *Find a leftmost derivation for this string.*
2. *Draw the derivation tree.*

Solution

1. Leftmost derivation is given below:

$$\begin{aligned}\dot{S} &\Rightarrow a\dot{S}\\ &\Rightarrow aa\dot{S}\\ &\Rightarrow aab\dot{S}S\\ &\Rightarrow aabb\dot{S}SS\\ &\Rightarrow aabba\dot{S}S\\ &\Rightarrow aabbaa\dot{S}\\ &\Rightarrow aabbaaa\end{aligned}$$

2. The derivation tree for the above string would be:



Exercise 4 *For the grammar given in Exercise 3, prove that every accepted word has more* `a`*'s than* `b`*'s using structural induction.*

Solution Let's say that a string is *green* when it has more `a`'s than `b`'s. We shall prove by induction that every string in the language is green. For this, we have to prove three things:

1. If S is green and S' is green then $\texttt{b}SS'$ is green.
2. If S is green then $\texttt{a}S$ is green.
3. `a` is green.

To prove (i), suppose S has x-many `a`'s and y-many `b`'s, and suppose S' has u-many `a`'s and v-many `b`'s. The

inductive hypothesis says $x > y$ and $u > v$. We deduce

$$\begin{array}{rcll} \text{number of } \mathtt{a}\text{'s in } \mathtt{b}SS' & = & & \\ x+u & \geqslant & (y+1)+(v+1) & \text{(since } x \geqslant y+1 \text{ and } u \geqslant v+1\text{)} \\ & = & y+v+2 & \\ & > & y+v+1 & \\ & = & \text{number of } \mathtt{b}\text{'s in } \mathtt{b}SS' & \end{array}$$

To prove (ii), suppose S has x $\mathtt{a}$'s and y $\mathtt{b}$'s. The inductive hypothesis says $x > y$. We deduce

$$\begin{array}{rcl} \text{number of } \mathtt{a}\text{'s in } \mathtt{a}S & = & x+1 \\ & > & x \\ & > & y \\ & = & \text{number of } \mathtt{b}\text{'s in } \mathtt{a}S \end{array}$$

We prove (iii) by

$$\begin{array}{rcl} \text{number of } \mathtt{a}\text{'s in } \mathtt{a}S & = & 1 \\ & > & 0 \\ & = & \text{number of } \mathtt{b}\text{'s in } \mathtt{a}S \end{array}$$

Exercise 5 *Let's look at a "Natural Language" example. The alphabet is*

{ *the, a, cat, dog, happy, tired, slept, died, ate, dinner, and, .* }

The grammar is

Sentence	$\Rightarrow$ *S*	::=	*C .*
Clause	*C*	::=	*NP VP* | *C and C*
Noun phrase	*NP*	::=	*Art N* | *dinner*
Noun	*N*	::=	*Adj N* | *cat* | *dog*
Adjective	*Adj*	::=	*happy* | *tired*
Verb phrase	*VP*	::=	*VI* | *VT NP*
Intransitive verb	*VI*	::=	*slept* | *died*
Transitive verb	*VT*	::=	*ate*
Article	*Art*	::=	*a* | *the*

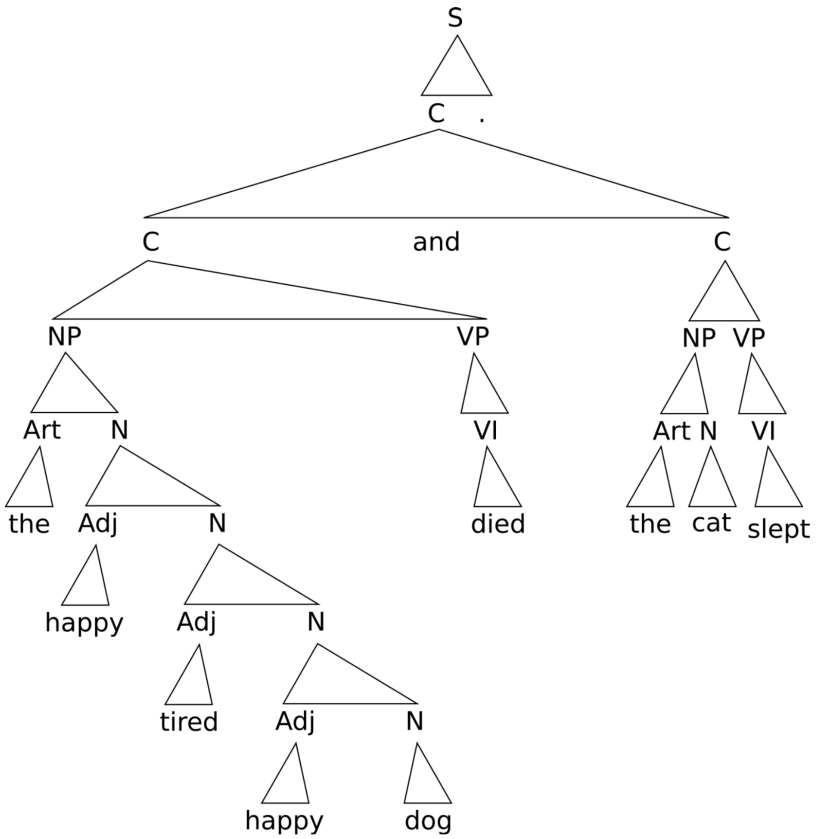
This grammar accepts "words" such as

the happy tired happy dog died and the cat slept.
the tired tired cat ate dinner.
dinner ate a happy dog.

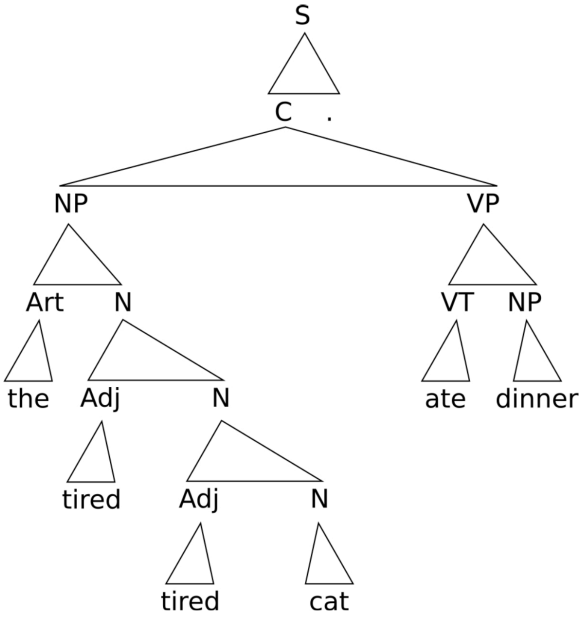
Try writing derivations and derivation trees for these sentences.

Solution

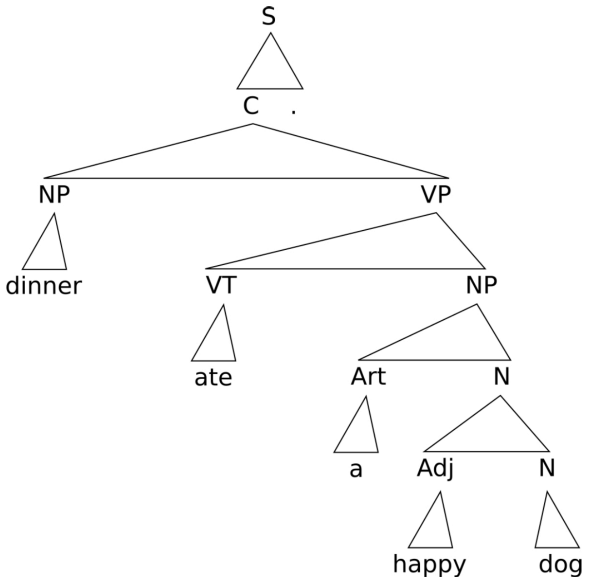
1. The derivation tree for "the happy tired happy dog died and the cat slept."



2. The derivation tree for “the tired tired cat ate dinner.”



3. The derivation tree for “dinner ate a happy dog.”



We will leave the derivations to you!

Exercise 6 *Show that the following grammar is ambiguous. The alphabet is* $\{\mathtt{a}, \mathtt{b}\}$.

$$\begin{array}{rcl} \Rightarrow P & ::= & \varepsilon \mid Q\mathtt{a} \mid \mathtt{a}Q \\ Q & ::= & \mathtt{aa}P \mid \mathtt{b}R \\ R & ::= & Q\mathtt{a} \end{array}$$

Solution

Here is the first leftmost derivation for the string “aaa”:

$$\begin{array}{rcl} \dot{P} & \Rightarrow & \dot{Q}a \\ & \Rightarrow & aa\dot{P}a \\ & \Rightarrow & aaa \end{array}$$

We can get the same “aaa” string using the following leftmost derivation:

$$\begin{array}{rcl} \dot{P} & \Rightarrow & a\dot{Q} \\ & \Rightarrow & aaa\dot{P} \\ & \Rightarrow & aaa \end{array}$$

Therefore, the given grammar is ambiguous!

Exercise 7 *Convert the following CFG into an equivalent CFG in Chomsky normal form*

$$\begin{array}{rcl} \Rightarrow A & ::= & BAB \mid B \mid \varepsilon \\ B & ::= & 00 \mid \varepsilon \end{array}$$

Solution

1. Add a new start variable S_0

$$\begin{aligned} \Rightarrow \boldsymbol{S_0} &\ \boldsymbol{::=} \quad \boldsymbol{A} \\ A &::= \quad BAB \mid B \mid \varepsilon \\ B &::= \quad 00 \mid \varepsilon \end{aligned}$$

2a. Remove ε-rule $B ::= \varepsilon$

$$\begin{aligned} \Rightarrow S_0 &::= \quad A \\ A &::= \quad B'AB' \mid B' \mid \boldsymbol{AB'} \mid \boldsymbol{B'A} \mid \boldsymbol{A} \mid \varepsilon \\ B' &::= \quad 00 \end{aligned}$$

2b. Remove ε-rule $A ::= \varepsilon$

$$\begin{aligned} \Rightarrow S_0 &::= \quad A' \mid \boldsymbol{\varepsilon} \\ A' &::= \quad B'A'B' \mid B' \mid A'B' \mid B'A' \mid A' \mid \boldsymbol{B'B'} \\ B' &::= \quad 00 \end{aligned}$$

Note: $S_0 ::= \varepsilon$ is allowed in CNF.

3a. Remove the unit rule $A' ::= A'$

$$\begin{aligned} \Rightarrow S_0 &::= \quad A' \mid \varepsilon \\ A' &::= \quad B'A'B' \mid B' \mid A'B' \mid B'A' \mid B'B' \\ B' &::= \quad 00 \end{aligned}$$

3b. Remove the unit rule $A' ::= B'$

$$\begin{aligned} \Rightarrow S_0 &::= \quad A' \mid \varepsilon \\ A' &::= \quad B'A''B' \mid \mathbf{00} \mid A'B' \mid B'A' \mid B'B' \\ B' &::= \quad 00 \end{aligned}$$

3c. Remove the unit rule $S_0 ::= A'$

$$\begin{aligned} \Rightarrow S_0 &::= \quad \boldsymbol{B'A'B'} \mid \mathbf{00} \mid \boldsymbol{A'B'} \mid \boldsymbol{B'A'} \mid \boldsymbol{B'B'} \mid \varepsilon \\ A' &::= \quad B'A'B' \mid 00 \mid A'B' \mid B'A' \mid B'B' \\ B' &::= \quad 00 \end{aligned}$$

4a. Convert the rules into proper form; introduce rule $D ::= 0$

$$\begin{aligned} \Rightarrow S_0 &::= \quad B'A'B' \mid \boldsymbol{DD} \mid A'B' \mid B'A \mid B'B' \mid \varepsilon \\ A' &::= \quad B'A'B' \mid \boldsymbol{DD} \mid A'B' \mid B'A' \mid B'B' \\ B' &::= \quad \boldsymbol{DD} \\ \boldsymbol{D} &\ \boldsymbol{::=} \quad \mathbf{0} \end{aligned}$$

4b. Convert the rules into proper form; introduce rule $C ::= A'B'$

$$\begin{aligned} \Rightarrow S_0 &::= \quad B'\mathbf{C} \mid DD \mid A'B' \mid B'A' \mid B'B' \mid \varepsilon \\ A' &::= \quad B'\mathbf{C} \mid DD \mid A'B' \mid B'A' \mid B'B' \\ B' &::= \quad DD \\ \boldsymbol{C} &\ \boldsymbol{::=} \quad \boldsymbol{A'B'} \\ D &::= \quad 0 \end{aligned}$$

The above CFG is now in Chomsky Normal Form.

Exercise 8 *Give grammars for the following two languages:*

1. *All binary strings with both an even number of zeroes and an even number of ones.*
2. *All strings of the form* $0^a1^b0^c$ *where* $a + c = b$*.*

Solution

1. All binary strings with both an even number of zeroes and an even number of ones.

$$\begin{aligned} \Rightarrow S &::= 0X \mid 1Y \mid \varepsilon \\ X &::= 0S \mid 1Z \\ Y &::= 1S \mid 0Z \\ Z &::= 0Y \mid 1X \end{aligned}$$

   The idea here is that:

   (a) Production S corresponds to having an even number of zeroes and an even number of ones.
   (b) Production X corresponds to having an odd number of zeroes and an even number of ones.
   (c) Production Y corresponds to having an even number of zeroes and an odd number of ones.
   (d) Production Z corresponds to having an odd number of zeroes and an odd number of ones.

   You can check this yourself by producing some binary strings using the grammar and keeping track of the parity of the number of zeroes and ones as you move through the productions.

2. All strings of the form $0^a1^b0^c$ where $a + c = b$.

$$\begin{aligned} \Rightarrow S &::= TU \\ T &::= 0T1 \mid \varepsilon \\ U &::= 1U0 \mid \varepsilon \end{aligned}$$

   This works because for every 0 generated on the right, we generate a 1 in the middle (see U) and for every 0 generated on the left, we generate another 1 in the middle (see T).