COMP30026 Models of Computation
Assignment 2

XuLin Yang, 904904

October 19, 2019

Challenge 1

Give context-free grammars for these languages:

a. The set A of odd-length strings in $\{\mathtt{a}, \mathtt{b}\}^*$ whose first, middle and last symbols are all the same. For example, b and ababa are in A, but ϵ, aaaa, and abbbb are not.

context-free grammar F is a 4-tuple (V, Σ, R, S) where
V is $\{T,\ X,\ Y\}$
Σ is $\{\mathtt{a},\ \mathtt{b}\}$
R is

$$\begin{array}{lcl} T & \to & \mathtt{a}\ X\ \mathtt{a} \mid \mathtt{b}\ Y\ \mathtt{b} \mid \mathtt{a} \mid \mathtt{b} \\ X & \to & \mathtt{a}\ X\ \mathtt{b} \mid \mathtt{b}\ X\ \mathtt{a} \mid \mathtt{b}\ X\ \mathtt{b} \mid \mathtt{a}\ X\ \mathtt{a} \mid \mathtt{a} \\ Y & \to & \mathtt{a}\ Y\ \mathtt{b} \mid \mathtt{b}\ Y\ \mathtt{a} \mid \mathtt{b}\ Y\ \mathtt{b} \mid \mathtt{a}\ Y\ \mathtt{a} \mid \mathtt{b} \end{array}$$

S is T

b. The set $B = \{\mathtt{a}^i\mathtt{b}\mathtt{a}^j \mid i \neq j\}$. For example, ab and abaaa are in B, but ϵ, a, b, and aabaa are not.

context-free grammar G is a 4-tuple (V, Σ, R, S) where
V is $\{T,\ X,\ Y,\ Z\}$
Σ is $\{\mathtt{a},\ \mathtt{b}\}$
R is

$$\begin{array}{lcl} T & \to & \mathtt{a}\ T\ \mathtt{a} \mid \mathtt{b}\ X \mid X\ \mathtt{b} \\ X & \to & \mathtt{a}\ X \mid \mathtt{a} \end{array}$$

S is T

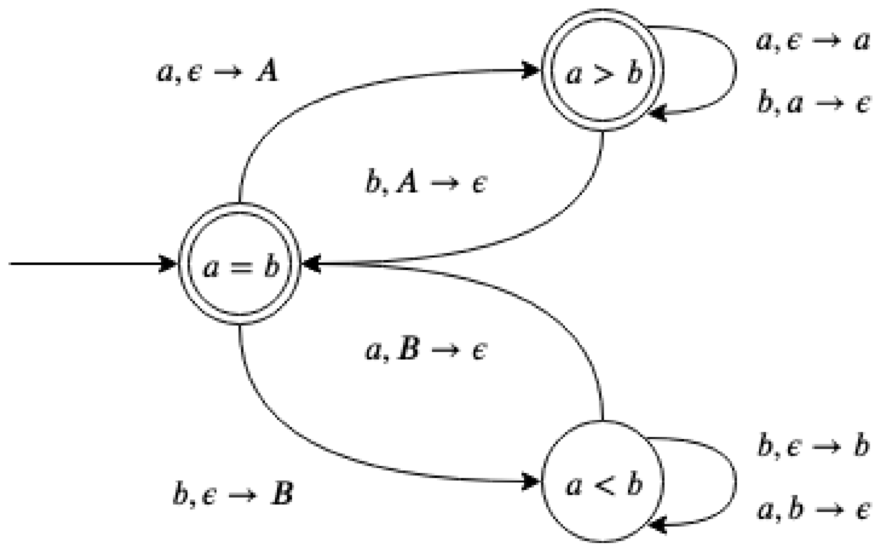
Challenge 2

Consider the language

$$C = \left\{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ contains at least as many as as bs}\right\}$$

For example, ϵ, aaa, aba, and bbaababaa are all in C, but bbb and bbaaabb are not.

a. Construct a 3-state push-down automaton to recognise C. Provide the solution as a transition diagram. Partial marks are given for a C recogniser with more than 3 states.



Note: could use '$' to replace 'A' and 'B' in the automaton. But since '$' means the start stack indicator in the subject, I use my 'A' and 'B' to be the indicator for start of a different state.

b. Prove formally that the following context-free grammar G generates C:

$$
\begin{aligned}
S &\rightarrow \epsilon \\
&\mid \texttt{a} \\
&\mid \texttt{a}\, S\, \texttt{b} \\
&\mid \texttt{b}\, S\, \texttt{a} \\
&\mid S\, S
\end{aligned}
$$

Hint: Proceed in two steps; prove that every string in $L(G)$ is in C (by structural induction) and prove that every string in C is in $L(G)$ (by induction on the length of the string).

Given:

context-free language C:

$$C = \{w \in \{\texttt{a}, \texttt{b}\}^* \mid w \text{ contains at least as many } \texttt{a}\text{s as } \texttt{b}\text{s}\}$$

and

context-free grammar G:

$$
\begin{aligned}
S &\rightarrow \epsilon \\
&\mid \texttt{a} \\
&\mid \texttt{a}\, S\, \texttt{b} \\
&\mid \texttt{b}\, S\, \texttt{a} \\
&\mid S\, S
\end{aligned}
$$

To be proved: G generates C

Proof:

(a) **Step 1**: prove that every string in $L(G)$ is in C (by structural induction)
Let string $x \in L(G)$, string $S' \in L(G)$, string $S'' \in L(G)$.
$A(string)$ = number of $\texttt{a}s$ in the $string$
$B(string)$ = number of $\texttt{b}s$ in the $string$

i. **base case:**

A. 1^{st} **base case:** $x = \epsilon$
As $\epsilon \in C$ by definition, so $x \in C$.
$\therefore$ 1^{st} base case is in C.

B. 2^{nd} **base case:** $x = \texttt{a}$
As $\texttt{a} \in C$ by definition, so $x \in C$.
$\therefore$ 2^{nd} base case is in C.

ii. **inductive case:**

A. 1^{st} **inductive case:** $x = \texttt{a}S'\texttt{b}$ where $S' \in L(G)$
As $S' \in C$, S' has at least as many $\texttt{a}$s as $\texttt{b}$s.
So add one $\texttt{a}$ at the start of S' and one $\texttt{b}$ at the end of S' still makes x has at least as many $\texttt{a}$s as $\texttt{b}$s.
In other words, as $A(x) = 1 + A(S')$, $B(x) = 1 + B(S')$ and $A(S') \geq B(S')$, so $A(x) \geq B(x)$
$\therefore$ 1^{st} inductive case is in C.

B. 2^{nd} **inductive case:** $x = \texttt{b}S'\texttt{a}$ where $S' \in L(G)$
Symmetrically, as $S' \in C$, S' has at least as many $\texttt{a}$s as $\texttt{b}$s.
So add one $\texttt{b}$ at the start of S' and one $\texttt{a}$ at the end of S' still makes x has at least as many $\texttt{a}$s as $\texttt{b}$s.
In other words, as $A(x) = 1 + A(S')$, $B(x) = 1 + B(S')$ and $A(S') \geq B(S')$, so $A(x) \geq B(x)$
$\therefore$ 2^{nd} inductive case is in C.

C. 3^{rd} **inductive case:** $x = S'S''$ where $S' \in L(G)$, $S'' \in L(G)$
As $A(x) = A(S') + A(S'')$, $B(x) = B(S') + B(S'')$, $A(S') \geq B(S')$, $A(S'') \geq B(S'')$, so $A(x) \geq B(x)$
$\therefore$ 3^{rd} inductive case is in C.

Hence, all of context free grammer rules in G generate string in C.
Therefore, every string in $L(G)$ is in C.

(b) **Step 2**: prove that every string in C is in $L(G)$ (by induction on the length of the string)
Let string $w \in C, |w| = \text{length}(w)$.
Assumption (Inductive hypothesis): If $|w| = n$ where $n \geq 0$, $w \in L(G)$. We can construct string of length $= |w| + 1$ or length $= |w| + 2$ that also holds by using the rules defined in $L(G)$.

i. **base case:**

A. 1^{st} **base case:** $n = 0$, $w = \epsilon$.
As $\epsilon \in L(G)$ by G's context free rule, so $w \in L(G)$.
$\therefore$ 1^{st} base case is in $L(G)$.

B. 2^{nd} **base case:** $n = 1$, $w = a$.
As $a \in L(G)$ by G's context free rule, so $w \in L(G)$.
$\therefore$ 2^{nd} base case is in $L(G)$.

i. **inductive case:**

A. 1^{st} **inductive case:** $\left|w'\right| = n+1$, $w' = aw$.
As $a \in L(G)$ by G's context free rule and $w \in L(G)$ by assumption, so $w' \in L(G)$ by G's context free rule: $S = S'S'', S \in L(G)$ where $S' \in L(G)$ and $S'' \in L(G)$.
$\therefore$ 1^{st} inductive case is in $L(G)$.

B. 2^{nd} **inductive case:** $\left|w'\right| = n+1$, $w' = wa$.
Symmetrically, as $a \in L(G)$ by definition and $w \in L(G)$ by assumption, so $w' \in L(G)$ by G's context free rule: $S = S'S'', S \in L(G)$ where $S' \in L(G)$ and $S'' \in L(G)$.
$\therefore$ 2^{nd} inductive case is in $L(G)$.

C. 3^{rd} **inductive case:** $\left|w'\right| = n+2$, $w' = awb$.
As $w \in L(G)$ by assumption, by G's context free rule: $S = aS'b$ where $S' \in L(G)$, therefore $S \in L(G)$.
So $w' \in L(G)$.
$\therefore$ 3^{rd} inductive case is in $L(G)$.

D. 4^{th} **inductive case:** $\left|w'\right| = n+2$, $w' = bwa$.
Symmetrically, as $w \in L(G)$ by assumption, by G's context free rule: $S = bS'a$ where $S' \in L(G)$, therefore $S \in L(G)$.
So $w' \in L(G)$.
$\therefore$ 4^{th} inductive case is in $L(G)$.

Hence, in $\left|w'\right| = n+1$, $\left|w'\right| = n+2$ inductive cases and base cases, the assumption if $|w| = n$ where $n \geq 0$, $w \in L(G)$ is true.
Therefore, every string in $L(G)$ is in C.

Conclusion: Since every string in $L(G)$ is in C and every string in C is in $L(G)$, we can say that context-free grammar G generates C.

Challenge 3

Consider the two language-transformer functions *triple* and *snip* defined as follows:

$$\begin{aligned} \mathit{triple}(L) &= \{www \mid w \in L\} \\ \mathit{snip}(L) &= \{xz \mid xyz \in L \text{ and } |x| = |y| = |z|\} \end{aligned}$$

Note that $\mathit{snip}(L)$ discards a string w from L unless w has length $3k$ for some $k \in \mathbb{N}$ (possibly 0), and then the strings whose lengths are multiples of 3 have their middle thirds removed. For example, if $L = \{\mathsf{ab}, \mathsf{bab}, \mathsf{bbb}, \mathsf{babba}, \mathsf{aabbaa}\}$ then $\mathit{snip}(L) = \{\mathsf{bb}, \mathsf{aaaa}\}$.

a. Let R be a regular language. Is $R^3 = R \circ R \circ R$ necessarily regular? Justify your answer.

   According to the theorem: The class of regular languages is closed under $\circ$.
   As R is regular, so does $R \circ R$, and so does $R \circ R \circ R$.
   $\therefore$ $R^3 = R \circ R \circ R$ is regular when R is regular.

b. Let R be a regular language. Is $\mathit{triple}(R)$ necessarily regular? Justify your answer.

   To be proved: $\mathit{triple}(R)$ is not necessarily regular, if R is regular.
   Let $R = \{a^*b\}$, R is regular.
   We have $\mathit{triple}(R) = \{a^iba^iba^ib \mid i \geq 0\}$.
   Pumping Lemma:
   L is regular $\Rightarrow \exists p \in \mathbb{N}_0\ \forall s \in L(\ |s| \geq p \Rightarrow \exists x, y, z(xyz = s \wedge\ |xy| \leq p \wedge\ |y| > 0 \wedge\ \forall i \in \mathbb{N}_0 |xy^iz \in L))$

Assume: $\mathit{triple}(R)$ is regular.
Let p be the pumping length, where $p \in \mathbb{N}_0$.
Pick $s = a^p b a^p b a^p b,\ s \in \mathit{triple}(R)$
$|s| = 3p + 3$ and $|s| \geq p$
Let x, y, z be strings
Assume $xyz = s,\ |xy| \leq p,\ |y| > 0$
Pick $i = 2$
$y = a^k,\ k > 0$ becuase $|xy| \leq p$ means y is made up of a

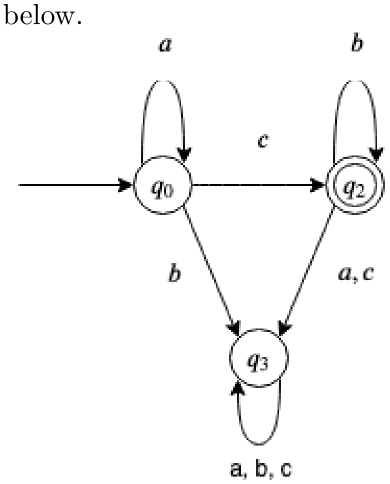
$$\begin{aligned} xy^iz &= xyyz & (1) \\ &= a^{p+k}ba^pba^pb & (2) \end{aligned}$$

$\therefore |xy^iz| \notin S$.
$\therefore$ $\mathit{triple}(R)$ is not regular.

In conclusion: $\mathit{triple}(R)$ is not necessarily regular if R is regular.

c. Let R be a regular language. Show that $\mathit{snip}(R)$ is not necessarily regular.

Let $R = \{a^*cb^* \cup \epsilon\}$, where $\Sigma = \{a, b, c, \epsilon\}$. And R is regular as there is a dfa that recognizes R given below.



Let S be a language, $S = \{a^nb^n | n \geq 0\}$, where $\Sigma = \{a, b, \epsilon\}$.
Let T be a regular language, $T = \{a^*b^*\}$, where $\Sigma = \{a, b, \epsilon\}$.

Assume: $\mathit{snip}(R)$ is regular.
And $\mathit{snip}(R) \cap T = S$. Because if c is not in $xz \in \mathit{snip}(R)$ where $xyz \in R$ means that there is enough number of as and bs near c such that when the middle part y is removed, the x is made up only of as and z is made up only of bs otherwise $xz \notin T$.
As $|x| = |z|$ and x, z are made up of a and b respectively, so $\mathit{snip}(R) \cap T = S$.

By theorem: The class of regular languages is closed under $\cap$. As $\mathit{snip}(R)$ and T are regular, so S is regular.
However, S is not regular.
Pumping Lemma:
L is regular $\Rightarrow \exists p \in \mathbb{N}_0\ \forall s \in L(\,|s| \geq p \Rightarrow \exists x, y, z(xyz = s \wedge |xy| \leq p \wedge |y| > 0 \wedge\ \forall i \in \mathbb{N}_0 | xy^iz \in L))$
Assume: S is regular.
Let p be the pumping length, where $p \in \mathbb{N}_0$.

Pick $s = a^p b^p,\ s \in S$
$|s| = 2p$ and $|s| \geq p$
Let x, y, z be strings
Assume $xyz = s$, $|xy| \leq p$, $|y| > 0$
Pick $i = 2$
$y = a^k,\ k > 0$ becuase $|xy| \leq p$ means y is made up of a

$$\begin{aligned} xy^i z &= xyyz && (3) \\ &= a^{p+k} b^p && (4) \end{aligned}$$

$\therefore \left|xy^i z\right| \notin S.$
$\therefore S$ is not regular.

As S is not regular, T is regular, by the theorem mentioned above, $snip(R)$ is not regular when R is regular.

In conclusion: $snip(R)$ is not necessarily regular if R is regular.