Solved selected problems of Introduction to the Theory of Computation by Michael Sipser.

Franco Zacco

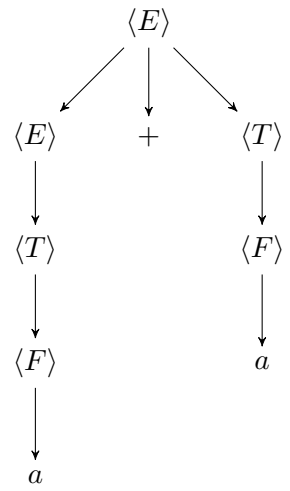
2 Context-Free Languages

2.1 Context-Free Grammars

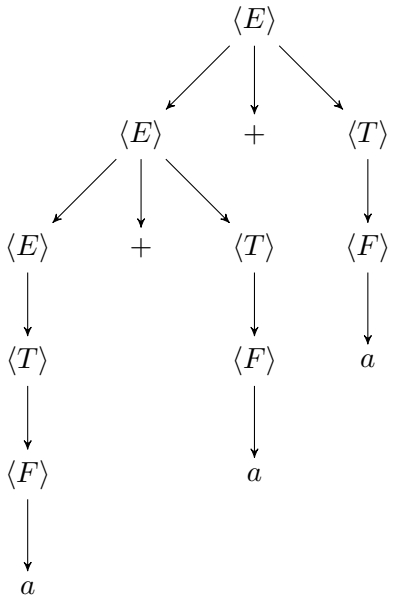
Proof. **2.1**

a. String: a

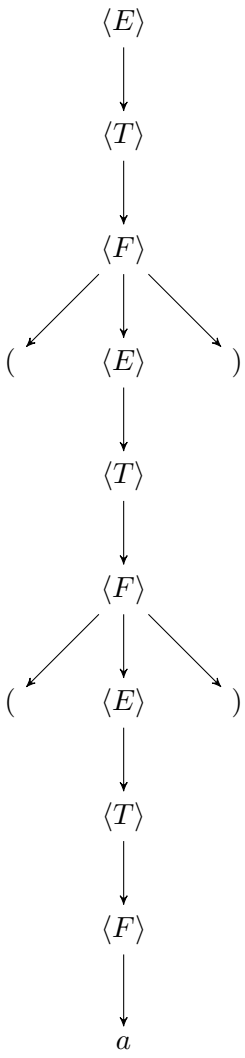
b. String: $a + a$



c. String: $a + a + a$



d. String: $((a))$



□

Proof. **2.2**

a. Let us consider the intersection of languages A and B we see that $C = A \cap B$ contains the strings for the case $m = n$ so

$$a^m b^n c^n = a^n b^n c^m$$

Then $C = \{a^n b^n c^n \mid n \geq 0\}$ but we saw in Example 2.36 that C is not a context-free language. Therefore the class of context-free languages is not closed under intersection.

b. Let us consider the complement of the union of languages A and B i.e. $\overline{A \cup B}$ then by DeMorgan's law we have that

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

But since the class of context-free languages is not closed under intersection then $\overline{A \cup B}$ might not be a context-free language. This implies that the class of context-free languages is not closed under complementation.

□

Proof. **2.3**

a. The variables of G are R, S, T and X.

b. The terminals of G are a, b and ε.

c. The start variable of G is R.

d. baa, ba and aab are three strings in $L(G)$.

e. aaa, bbb and ε are three strings not in $L(G)$.

f. False because T does not yield aba but $XTX \mid X \mid \varepsilon$.

g. True because $T \Rightarrow XTX \Rightarrow aTX \Rightarrow aTa \Rightarrow aXb \Rightarrow aba$.

h. False because T does not yield T but $XTX \mid X \mid \varepsilon$.

i. False because there are no finitely many steps such that T can be derived into T.

j. True because $XXX \Rightarrow aXX \Rightarrow abX \Rightarrow aba$.

k. False because X derives into $a \mid b$.

l. True because $T \Rightarrow XTX \Rightarrow X\varepsilon X \Rightarrow XX$.

m. True because $T \Rightarrow XTX \Rightarrow XXX$.

n. False because $S \Rightarrow aTb \mid bTa$ and there is no way to remove the a's or b's after that.

o. $L(G)$ is the language of strings which always contain a's and b's of length at least 2 i.e. a, aa, bb, aaa, $bbbb$, etc. are not in the language.

□

Proof. **2.4**

a. $\{w \mid w \text{ contains at least three 1s}\}$

$$\begin{aligned} R_1 &\to 0R_1 \mid 1R_2 \\ R_2 &\to 0R_2 \mid 1R_3 \\ R_3 &\to 0R_3 \mid 1R_4 \\ R_4 &\to 0R_4 \mid 1R_4 \\ R_4 &\to \varepsilon \end{aligned}$$

b. $\{w \mid w \text{ starts and ends with the same symbol}\}$

$$\begin{aligned} R_1 &\to 0R_20 \mid 1R_21 \\ R_2 &\to 0R_2 \mid 1R_2 \mid \varepsilon \end{aligned}$$

c. $\{w \mid \text{the length of } w \text{ is odd}\}$

$$\begin{aligned} R_1 &\to 0R_2 \mid 1R_2 \\ R_2 &\to 0R_1 \mid 1R_1 \mid \varepsilon \end{aligned}$$

d. $\{w \mid \text{the length of } w \text{ is odd and its middle symbol is a } 0\}$

$$R_1 \to 0R_10 \mid 0R_11 \mid 1R_10 \mid 1R_11 \mid 0$$

e. $\{w \mid w = w^R, \text{ that is, } w \text{ is a palindrome}\}$

$$R_1 \to 0R_10 \mid 1R_11 \mid 0 \mid 1$$

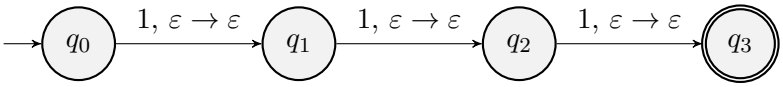
f. The empty set.

$$R_1 \to R_1$$

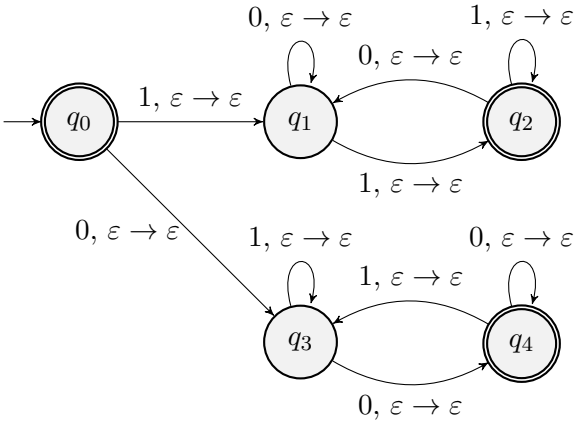
□

Proof. **2.5**

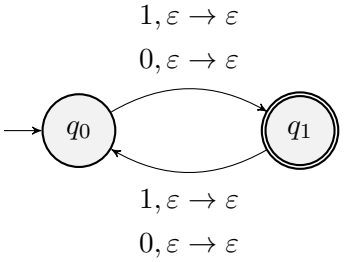
a. $\{w \mid w \text{ contains at least three 1s}\}$



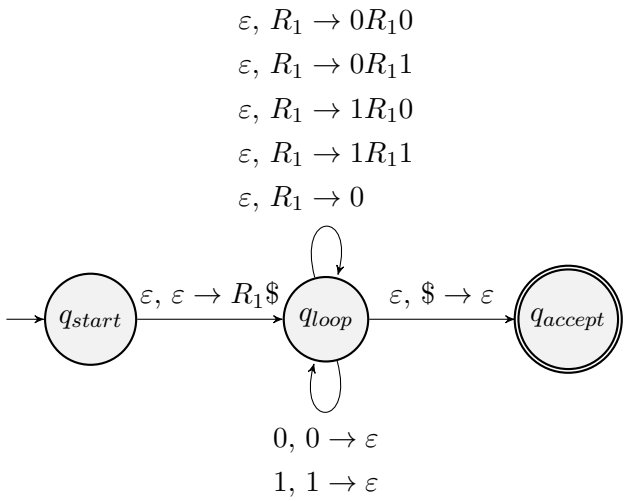
b. $\{w \mid w \text{ starts and ends with the same symbol}\}$



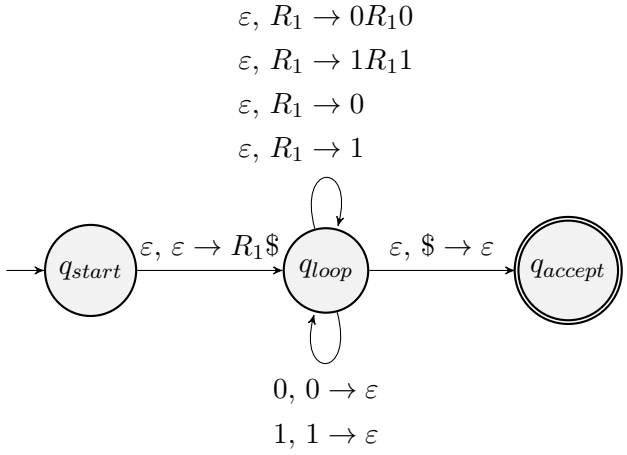
c. $\{w \mid \text{the length of } w \text{ is odd}\}$



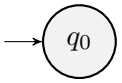
d. $\{w \mid$ the length of w is odd and its middle symbol is a $0\}$



e. $\{w \mid w = w^R$, that is, w is a palindrome$\}$



f. The empty set.



□

Proof. **2.6**

a. The set of strings over the alphabet $\{a, b\}$ with more a's than b's

$$R_1 \to bR_1a \mid aR_1b \mid baR_1 \mid R_1ba \mid abR_1 \mid R_1ab \mid R_2$$
$$R_2 \to R_1 \mid a$$

b. The complement of the language $\{a^n b^n \mid n \geq 0\}$

$$R_1 \to bR_1a \mid baR_1 \mid R_1ba \mid abR_1 \mid R_1ab \mid R_2$$
$$R_2 \to R_1 \mid a \mid b$$

c. $\{w\#x \mid w^R$ is a substring of x for $w, x \in \{0,1\}^*\}$

$$R_0 \to R_1R_2$$
$$R_1 \to 0R_10 \mid 1R_11 \mid \#R_2$$
$$R_2 \to 0R_2 \mid 1R_2 \mid \varepsilon$$

d. $\{x_1\#x_2\#...\#x_k \mid k \geq 1$, each $x_i \in \{a, b\}^*$, and for some i and j, $x_i = x_j^R\}$

$$R_0 \to R_1 \mid R_2\#R_1$$
$$R_1 \to aR_1 \mid bR_1 \mid \#R_1 \mid \varepsilon \mid R_0$$
$$R_2 \to aR_2a \mid bR_2b \mid \varepsilon \mid R_1$$

□

Proof. **2.7**

a. The set of strings over the alphabet $\{a, b\}$ with more a's than b's

The PDA for this language should do as follows, we read the first symbol, suppose it's an a then we push it to the stack, then we read the next symbol if it's a b we pop the stack, if it's another a we push it to the stack. If the first symbol was a b and the next one is an a then we pop the b and so on. We continue this process until we read the entire input. If we only have a's in the stack, we accept it; otherwise, we reject it.

b. The complement of the language $\{a^n b^n \mid n \geq 0\}$

The PDA for this language should do as follows.
If the first input symbol is a b we accept.
If the first input string is a set of a's we push them to the stack and if the next symbols are b's we pop the stack for each one of them to check if we have the same number of a's. If that's the case, we reject otherwise, we accept.
If after any number of a's and b's we still read another a we accept.

c. $\{w\#x \mid w^R$ is a substring of x for $w, x \in \{0, 1\}^*\}$

The PDA for this language should do as follows.
We read each symbol and we push it to the stack until we read a #, if there is no # we reject. Then non-deterministically we read the subsequent symbols and check if they match to the ones in the stack, if they match, we pop them from the stack. If the stack is empty in any of the non-deterministic branches when we finish reading the input we accept. Otherwise we reject.

d. $\{x_1\#x_2\#...\#x_k \mid k \geq 1$, each $x_i \in \{a, b\}^*$, and for some i and j, $x_i = x_j^R\}$

The PDA for this language should do as follows.
Suppose $x_2 = x_1^R$, then we read each symbol of x_1 and we push it to the stack until we read a # then we read x_2 and we pop the symbols that match with the ones we have on the stack since $x_2 = x_1^R$.
Now suppose $x_j = x_1^R$ then we do the same but non-deterministically, we check each x_j with the process mentioned in the previous paragraph.
Finally, if x_1 is not the string reversed, then we have to check non-deterministically every x_i as we did before to check which one is the one reversed. In each non-deterministic branch, we store x_i in the sack, and we check as in the previous paragraph which x_j matches to x_i^R.
In the case the input has no # or no string matches to another reversed string we reject.

□

Proof. **2.8** One leftmost derivation of "the girl touches the boy with the flower" in grammar G_2 is

$$\begin{aligned}
\langle SENTENCE\rangle &\Rightarrow \langle NOUN-PHRASE\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \langle CMPLX-NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \langle ARTICLE\rangle\langle NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the } \langle NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the girl } \langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the girl } \langle CMPLX-VERB\rangle \\
&\Rightarrow \text{the girl } \langle VERB\rangle\langle NOUN-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle NOUN-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle CMPLX-NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle ARTICLE\rangle\langle NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the } \langle NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the boy } \langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the boy } \langle PREP\rangle\langle CMPLX-NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with } \langle CMPLX-NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with } \langle ARTICLE\rangle\langle NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with the } \langle NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with the flower}
\end{aligned}$$

But we also have the following leftmost derivation of this phrase

$$\begin{aligned}
\langle SENTENCE\rangle &\Rightarrow \langle NOUN-PHRASE\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \langle CMPLX-NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \langle ARTICLE\rangle\langle NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the } \langle NOUN\rangle\langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the girl } \langle VERB-PHRASE\rangle \\
&\Rightarrow \text{the girl } \langle CMPLX-VERB\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl } \langle VERB\rangle\langle NOUN-PHRASE\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle NOUN-PHRASE\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle CMPLX-NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches } \langle ARTICLE\rangle\langle NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the } \langle NOUN\rangle\langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the boy } \langle PREP-PHRASE\rangle \\
&\Rightarrow \text{the girl touches the boy } \langle PREP\rangle\langle CMPLX-NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with } \langle CMPLX-NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with } \langle ARTICLE\rangle\langle NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with the } \langle NOUN\rangle \\
&\Rightarrow \text{the girl touches the boy with the flower}
\end{aligned}$$

□

Proof. **2.13**

a. $L(G)$ describes strings with two #s betwen a number of 0s including no 0s i.e. two # between an even number of 0s.

Also, $L(G)$ describes strings starting with n-zeros a # and ending with twice as many zeros ($2n$ zeros).

b. Suppose $L(G)$ is a regular language, we want to arrive at a contradiction.

Let $s = 0^p\#0^{2p}$ we see that $s \in L(G)$ then by the pumping lemma s can be divided into three pieces $s = xyz$ such that $|y| > 0$ and $|xy| \leq p$ but also has to be that $xy^iz \in L(G)$ for each $i \geq 0$.

Because we need that $|xy| \leq p$ then the only way we can divide s is as follows.

Let $0 \leq j < k \leq p$ and let us divide s as

$$\underbrace{0^j}_{x}\underbrace{0^k}_{y}\underbrace{0^{p-k}\#0^{2p}}_{z}$$

We see that $|y| > 0$ and $|xy| \leq p$ as desired. Now, if we take $i = 2$ should be that $0^j0^{2k}0^{p-k}\#0^{2p}$ is in $L(G)$ but for this to happen must be that

$$\begin{aligned} 2(j + 2k + (p - k)) &= 2p \\ 2j + 2k &= 0 \\ k &= -j \end{aligned}$$

Which cannot happen and hence s does not comply with the pumping lemma.

Therefore we have arrived at a contradiction and must be that $L(G)$ is not regular.

□

Proof. **2.16** Let $L(G_1)$ and $L(G_2)$ be context-free languages generated by the context-free grammars $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ then we can construct

$$G_U = (V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{r\}, S)$$

where r sets the new start variable such that $S \to S_1 \mid S_2$. Then $L(G_U) = L(G_1) \cup L(G_2)$ and hence the context-free languages are closed under the union operation.

Now, let us contruct

$$G_\circ = (V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{r\}, S)$$

but in this case we define r as $S \to S_1 S_2$ so $L(G_\circ) = L(G_1) \cap L(G_2)$ and hence the context-free languages are closed under the concatenation operation.

Finally, we construct

$$G_* = (V_1 \cup \{S\}, \Sigma, R_1 \cup \{r\}, S)$$

where in this case we define r as $S \to SS_1|\varepsilon$ so $L(G_*) = L(G_1)^*$ and hence the context-free languages are closed under the star operation. □

Proof. **2.17** Let R be a regular expression, then R is

1. a for some a in the alphabet Σ
2. ε
3. $\emptyset$
4. $R_1 \cup R_2$ where R_1 and R_2 are regular expressions
5. $R_1 \circ R_2$ where R_1 and R_2 are regular expressions
6. R_1^* where R_1 is a regular expression

Then if $R = a$ we can define a context-free grammar G such that

$$S \to a$$

Which recognizes a. If $R = \varepsilon$ then we define G as

$$S \to \varepsilon$$

And if $R = \emptyset$ then we define G as

$$S \to S$$

In any of the rest of the cases, context-free languages are closed under union, concatenation, and star operation, so we can always find a context-free grammar that is equivalent to the regular expression R. Therefore, every regular language is context-free. □

Proof. **2.30**

a. Let $A = \{0^n1^n0^n1^n \mid n \geq 0\}$, we want to show that A is not context free by contradiction, so let us assume that A is context-free.

Let us select a string $s = 0^p1^p0^p1^p$ which is a member of A and of length at least p, then s can be divided into $s = uvxyz$.

Condition 2 stipulates that either v or y is nonempty. Then we consider two cases.

1. Suppose v and y contains only one type of alphabet symbol, then pumping s to uv^2xy^2z will have different number of 0s and/or 1s in the string and so uv^2xy^2z will not be in A. Hence we have a contradiction.
2. When either v or y contain more than one type of symbol then uv^2xy^2z may contain equal numbers of 0s and 1s but not in the correct order. Hence we have another contradiction.

Because both cases result in a contradiction, a contradiction is unavoidable. Therefore A is not context free.

b. Let $A = \{0^n\#0^{2n}\#0^{3n} \mid n \geq 0\}$, we want to show that A is not context free by contradiction, so let us assume that A is context-free.

Let us select a string $s = 0^p\#0^{2p}\#0^{3p}$ which is a member of A and of length at least p, then s can be divided into $s = uvxyz$.

Condition 2 stipulates that either v or y is nonempty. Then we consider two cases.

1. Suppose v and y contains only one type of alphabet symbol, then pumping s to uv^2xy^2z creates a string which either has two #s if v or y equals to # or we have a string that doesn't match the expected form i.e. has more 0s in one part of the string, so the lengths $n, 2n, 3n$ of each substring is not maintained. Hence we have a contradiction.
2. When either v or y contain more than one type of symbol then uv^2xy^2z creates a string with two # and so uv^2xy^2z is not in A. Hence we have another contradiction.

Because both cases result in a contradiction, a contradiction is unavoidable. Therefore A is not context free.

c. Let $A = \{w\#t \mid w \text{ is a substring of } t, \text{ where } w, t \in \{a, b\}^*\}$ we want to show that A is not context free by contradiction, so let us assume that A is context-free.

Let us select the string $s = a^p b^p \# a^p b^p$ which is a member of A and of length at least p, then s can be divided into $s = uvxyz$.

Condition 2 stipulates that either v or y is nonempty. Then we consider two cases.

1. Suppose v and y contains only one type of alphabet symbol and not #, then three cases can happen
   i. If v and y are on the left side of the string (to the left of the #) then the left side of the pumped string uv^2xy^2z is not a substring of the right side of the string. A contradiction.
   ii. If v and y are on the right side of the string then the "down pumped" string uv^0xy^0z creates a string into which the left side is not a substring of the right side. A contradiction.
   iii. Finally, if v is on the left side and y is on the right side then v has to be formed by some number of b's and y has to be some number of a's, otherwise, we have that $|vxy| > p$. Then the pumped string uv^2xy^2z is not in A since the left side is not a substring of the right side (it has more b's). A contradiction.
2. When either v or y contain more than one type of symbol then two cases can happen
   i. If v and y are on the left side of the string then the pumped string uv^2xy^2z is not in A because the left side of the string is not a substring of the right side. A contradiction.
   i. If v and y are on the right side of the string then the "down pumped" string uv^0xy^0z is not in A since the left side is not a substring of the right side. A contradiction.

Because all cases result in a contradiction, a contradiction is unavoidable. Therefore A is not context free.

d. Let

$$A = \{t_1\#t_2\#...\#t_k \mid k \geq 2, \text{ each } t_i \in \{a,b\}^*, \text{ and } t_i = t_j \text{ for some } i \neq j\}$$

we want to show that A is not context free by contradiction, so let us assume that A is context-free.

Let $t_1 = t_2 = a^p b^p$ then the string $s = a^p b^p \# a^p b^p$ is in A and of length at least p, then s can be divided into $s = uvxyz$.

As we showed before if v and y are on the left side of the string (left to the $\#$) then the left side of uv^2xy^2z is going to be different from the right side and hence $t_1 \neq t_2$.

The same can be shown for v and y on the right side of the string.

So the only option left is that v is on the left side and y is on the right side but then v has to be formed by some number of b's and y has to be some number of a's, otherwise, we have that $|vxy| > p$.

Then the pumped string uv^2xy^2z is not in A since the left side is not equal the right side.

Therefore in any case we get a contradiction and hence A is not context free.

□