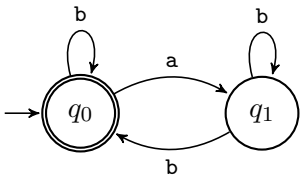
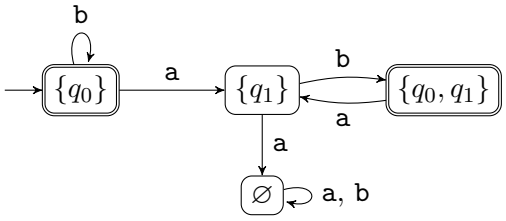
## Problem 1 (30 points)

Consider the following NFA $M$:




(a) (10 points) Convert $M$ to a DFA.

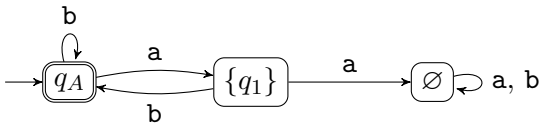



(b) (10 points) Run the DFA minimization algorithm on the resulting DFA. Prove that every pair of states is distinguishable.

Initially, we mark the pairs of states where one is accepting and the other rejecting as distinguishable. Then we see that $q_\varnothing$ and $q_1$ are distinguishable on input $b$. This only leaves the pair $(q_0, q_{01})$. These cannot be distinguished by either $a$ or $b$, so we group them into state $q_A$.

| | | | |
|---|---|---|---|
| $q_0$ | $\times$ | | |
| $q_1$ | $\times$ | $\times$ | |
| $q_{01}$ | $\times$ | $A$ | $\times$ |
| | $q_\varnothing$ | $q_0$ | $q_1$ |

This yields the following 3-state DFA:




The pair $(q_A, q_1)$ is distinguishable as one is accepting the other rejecting. The pair $(q_A, q_\varnothing)$ is distinguishable for the same reason. Finally, $(q_\varnothing, q_1)$ is distinguishable because on input $b$, $q_1$ goes to an accepting state while $q_\varnothing$ goes to a rejecting state.

(c) (10 points) Write a regular expression for the language of $M$.

One can see from the minimized DFA that any string containing `aa` will be rejected, and so is any string ending in `a`. It is natural to guess that any other string will be accepted. A string not containing any `aa` or ending in an `a` can be broken into blocks, where each block is either a `b` or `ab`. Indeed, any string will be accepted by the minimized DFA, because starting from the initial state $q_A$, upon reading `b` or `ab` brings us back to $q_A$, which is accepting.

Therefore the desired regular expression is $(\mathtt{b}|\mathtt{ab})^*$.

## Problem 2 (20 points)

Let $L$ be the language $\{\mathtt{a}^n\mathtt{b}^n \mid n \geqslant 0\}$ over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) (10 points) Write a context-free grammar for $L$. Your CFG should be unambiguous and in Chomsky Normal Form.

The language $L$ is that it contains the empty string, which has to be treated specially in Chomsky Normal Form grammars. If we take out the empty string from $L$, we can describe $L$ by the CFG $S \to \mathtt{a}S\mathtt{b} \mid \mathtt{ab}$. After converting to Chomsky Normal Form, we obtain the CFG $S \to AT \mid AB, T \to SB, A \to \mathtt{a}, B \to \mathtt{b}$. We now add the special rule $S \to \varepsilon$ to obtain the CFG

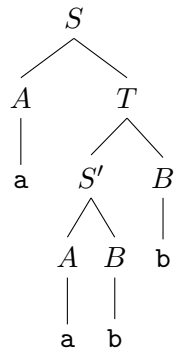
$$\begin{aligned} S &\to AT \mid AB \mid \varepsilon \\ T &\to SB \\ A &\to \mathtt{a} \\ B &\to \mathtt{b}. \end{aligned}$$

Technically, this CFG is ambiguous owing to the special production $S \to \varepsilon$ and because $S$ appears on the right hand side of a rule. To obtain a truly unambiguous CFG, we can introduce an additional variable $S'$ that derives the same strings as $S$ except $\varepsilon$:

$$\begin{aligned} S &\to AT \mid AB \mid \varepsilon \\ S' &\to AT \mid AB \\ T &\to S'B \\ A &\to \mathtt{a} \\ B &\to \mathtt{b}. \end{aligned}$$

(b) (10 points) Using the CFG from part (a), apply the Cocke–Younger–Kasami algorithm on input `aabb`. Draw the parse tree derived by the algorithm.

| | | | |
|---|---|---|---|
| $S\|S'$ | | | |
| – | $T$ | – | |
| – | $S\|S'$ | – | |
| $A$ | $A$ | $B$ | $B$ |
| `a` | `a` | `b` | `b` |



## Problem 3 (30 points)

Consider the following context-free grammar $G$:

$$S \to SS \mid \texttt{aa}S\texttt{bb} \mid \varepsilon$$

(a) (10 points) For each of these strings, say if it is in the language of $G$. Justify your answer.

`aaaabbbbaabb`: This string is in the language of $G$ as it has the following derivation:

$$S \Rightarrow SS \Rightarrow \texttt{aa}S\texttt{bb}S \Rightarrow \texttt{aaaa}S\texttt{bbbb}S \Rightarrow \texttt{aaaabbbb}S \Rightarrow \texttt{aaaabbbbaa}S\texttt{bb} \Rightarrow \texttt{aaaabbbbaabb}.$$

`aaab`: Not in the language of $G$. According to the productions of $G$, every string in $L(G)$ should contain equal even number of `a`'s, so `aaab` is not in $L(G)$.

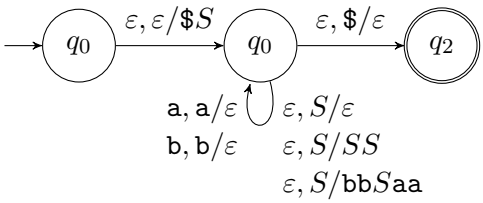
`abaaaabbbb`: Not in the language of $G$. According to the productions of $G$, every appearance of `a` must be followed by another `a`, and here the first `a` is not.

(b) (10 points) Show that $G$ is ambiguous.

For instance, $\varepsilon \in L(G)$ has two left-most derivations: $S \Rightarrow SS \Rightarrow S \Rightarrow \varepsilon$ and $S \Rightarrow \varepsilon$.

(c) (10 points) Draw a pushdown automaton for the language of $G$. Specify all the states, transitions, and start/final states.

There are several ways to do this. One way is to apply the generic CFG to PDA transformation from class:




## Problem 4 (20 points)

For each of the following languages, say whether it is (i) regular, (ii) context-free but not regular, or (iii) not context-free. Justify your answer by describing a DFA, NFA, regular expression, PDA, CFG, and/or giving a proof via the pumping lemma or pairwise distinguishable strings.

(a) (10 points) $L_1 = \{\texttt{a}^i\texttt{b}^j\texttt{c}^k \mid j = i + k \text{ and } i, j, k \geqslant 0\}$

Please circle: **regular** **context-free but not regular** **not context-free**

To show $L_1$ is context-free, we design a CFG for it. The idea is to match the `a`s and the `b`s on the left, and then the `b`s and the `c`s on the right:

$$S \to LR$$
$$L \to \texttt{a}L\texttt{b} \mid \varepsilon$$
$$R \to \texttt{b}R\texttt{c} \mid \varepsilon.$$

To show $L_1$ is not regular, we apply the pumping lemma for regular languages. If $L_1$ was regular, then for some $n$, every sufficiently long string $x \in L_1$ of length $n$ or more can be written as $x = uvw$ so that $|uv| < n$, $v \neq \varepsilon$ and $uv^i w \in L_1$ for every $i \geqslant 0$. Let $x = \mathtt{a}^n\mathtt{b}^n$. If $x = uvw$ where $|uv| < n$ and $v \neq \varepsilon$, then $uv^2w = \mathtt{a}^m\mathtt{b}^n$ where $m > n$, so it is not in $L_1$. So $L_1$ cannot be regular.

(b) (10 points) $L_2 = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid$ every block of $\mathtt{a}$'s in $w$ contains exactly three $\mathtt{a}$'s$\}$

Please circle: **regular** **context-free but not regular** **not context-free**

$L_2$ is regular. For instance the expression $\mathtt{b}^*\mathtt{(aaabb}^*\mathtt{)}^*\mathtt{(}\varepsilon\mathtt{+aaa)}$ is a regular expression for $L_2$. The following NFA also accepts $L_2$: