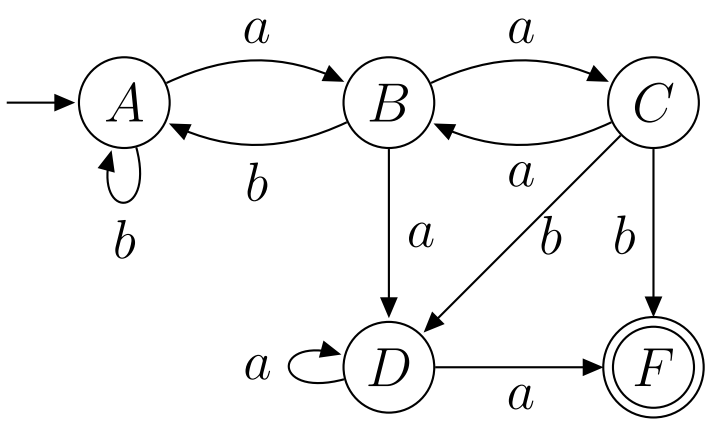
Test 2

1. Construct a nondeterministic finite automaton from the following grammar.

$$\begin{aligned} A &\longrightarrow \mathtt{a}B \mid \mathtt{b}A \\ B &\longrightarrow \mathtt{a}C \mid \mathtt{a}D \mid \mathtt{b}A \\ C &\longrightarrow \mathtt{a}B \mid \mathtt{b}D \mid \mathtt{b} \\ D &\longrightarrow \mathtt{a}D \mid \mathtt{a} \end{aligned}$$



(States correspond to variables and the new accept state F. The start state is the start variable. The arrows go according to the grammatical rules. When there is no variable on the right hand side then the arrow points to F.)

Neptun: **Name:** SOLUTION

2. Let A and B two context-free languages.

a) Does it follow that their union $A \cup B$ is also context-free? Prove your claim.

Yes.
A and B have CF grammars (V_A, Σ, R_A, S_A) and $(V_B, \Sigma, R_B S_B)$. We can assume that $V_A \cap V_B = \emptyset$. A grammar (V, Σ, R, S) for $A \cup B$ has $V = V_A \cup V_B \cup \{S\}$ where S is the new start variable. The rules are $R_A \cup R_B$ and the rules $S \longrightarrow S_A \mid S_B$. To obtain CF grammar it might be necessary to get rid of the ε rules (when $\varepsilon \in A \cup B$).

b) Does it follow that their intersection $A \cap B$ is also context-free? (You do not have to prove this.)

No.

c) Does it follow that the complement language $\overline{A}$ is also context-free? Prove your claim.

No. Assume it is. Then for any CF languages A and B, their complements $\overline{A}$ and $\overline{B}$ are CF. Then their union $\overline{A} \cup \overline{B}$ is also CF. By the assumption, its complement $\overline{\overline{A} \cup \overline{B}}$ is CF. Since $\overline{\overline{A} \cup \overline{B}} = A \cap B$, this would imply that the CF languages are closed under intersection but that is not true (see b).

3. Consider the pushdown automaton $M = (Q,\Sigma,\Gamma,\delta,q_0,F)$, where $Q = \{q_0,q_1,q_a,q_b,q_c,q_f\}$, $\Sigma = \{a,b,c\}$, $\Gamma = \{A,\$\}$, $F = \{q_f\}$, and the transition function δ is the following

$$\begin{array}{lcl@{\qquad\qquad}lcl}
\delta(q_0,\varepsilon,\varepsilon) & = & (q_1,\$) & \delta(q_b,b,A) & = & (q_b,\varepsilon) \\
\delta(q_1,a,\$) & = & (q_a,A\$) & \delta(q_b,\varepsilon,A) & = & (q_f,A) \\
\delta(q_a,a,A) & = & (q_a,AA) & \delta(q_c,c,A) & = & (q_c,\varepsilon) \\
\delta(q_a,b,A) & = & (q_b,\varepsilon) & \delta(q_c,c,\$) & = & (q_f,\$) \\
\delta(q_a,c,A) & = & (q_c,\varepsilon) & \delta(q_f,c,\$) & = & (q_f,\$)
\end{array}$$

(a) Describe the language $L(M)$ accepted by M.

$L(M) = \{a^ib^k : i > k \geq 0\} \cup \{a^ic^m : 1 \leq i < m\}$

(b) Explain why the strings of $L(M)$ are accepted.

We have to show an accepting path for each of these strings.

The states ensure that in L every string is in either a^*b^* or a^*c^*. For $w = a^ib^k$ (after pushing a \$) in state q_a an A is pushed for each a, then in state q_b an A is popped for each b. To move to the accept state q_f after the last b there has to be an A on the top of the stack. This is true when there are more a's than b's. Similarly for $w = a^ic^m$ after pushing an A for each a the machine is in state q_c where it pops one A for each c. To move from q_c to q_f the stack has to be empty (except the $\$$), so there are more c's than a's. The extra c's can be read in q_f.

(c) Is this pushdown automaton deterministic? Why?

No because of the first two transition of the second column: at q_b with A on the top of the stack there are two possibilities: either read an a from the input or move to q_f without reading.

4. Prove that the following language is not context-free. (Use the pumping lemma.)

$$L = \{a^i b^j c^k : |i-j| = 2 \text{ and } |j-k| = 2\}$$

where $|i-j|$ and $|j-k|$ denotes the absolute value of the differences $i-j$ and $j-k$, respectively. (Examples: $ab^3c^5 \in L$, $a^6b^8c^6 \in L$, $a^4b^2 \in L$.)

Assume it is CF. The pumping lemma can be applied, let p be the pumping length and let $w = a^p b^{p+2} c^p$. Then $w \in L$ and $|w| = 3p+2 > p$, so by the lemma it has a form $w = uvxyz$, where $|vxy| \le p$, $|vy| \ge 1$ and $uv^kxy^kz \in L$ for each k.

- If v (or y) is not homogeneous (contains different letters) and $k > 1$ then in the string uv^kxy^kz the letters are not in the right order, $uv^kxy^kz \notin L$.
- v and y have only a's. Choose $k = 0$: $\#a < p$, $\#b = p+2$, so $uxz \notin L$.
  ($k = 2$ is not good when $|vy| = 4$ because $a^{p+4}b^{p+2}c^p \in L$, but, for example, $k = 5$ works since it adds at least $k \cdot |vy| > 4$ a's to w.)
- v has only a's, y only b's (can assume they are not empty). Choose $k = 2$: $\#b > p+2$ but $\#c = p$ so $uv^2xy^2z \notin L$.
- v has only a's, y only c's: not allowed by the lemma, since $|vxy| > p$.
- v and y have only b's: Choose $k = 2$: there will be too many b's.
- v has only b's, y only c's . This is like the case of a's and b's.
- v and y have only c's. This is like the case of a's.

(Other choices for w can be also good,)