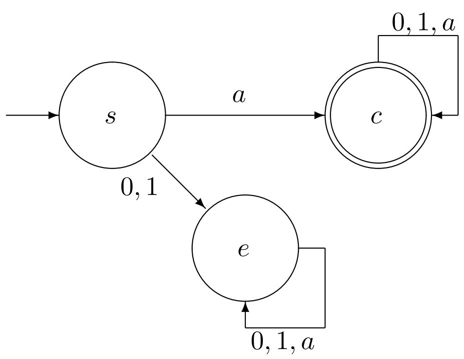
# Solutions to Homework 3

**Task:** Apply the general algorithm for transforming the finite automaton into a regular language (i.e., a language described by a regular expression) to Automaton B from Problem 1.4. For simplicity, assume that we only have symbols 0, 1, and $a$. Eliminate first the error state, then the start state, and finally, the state $c$.

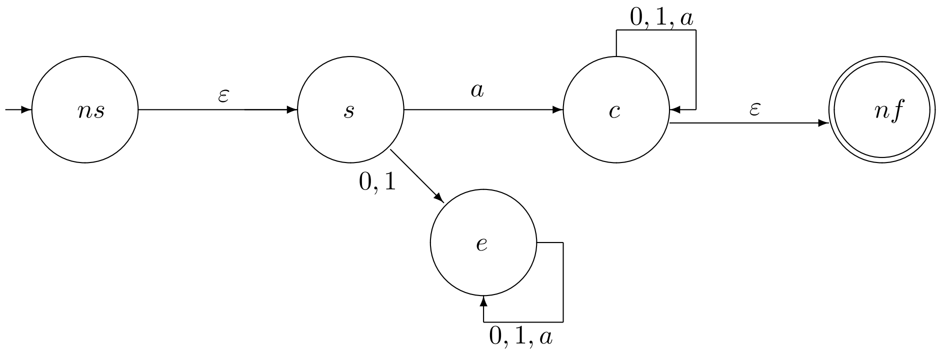
**Solution.** We start with the described automaton:



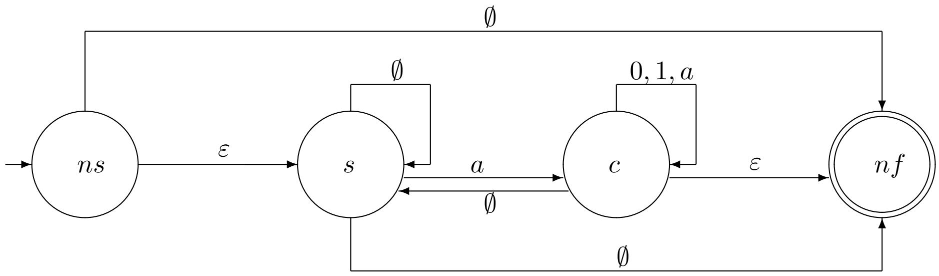

According to the general algorithm, first we add a new start state $ns$ and a few final state $nf$, and we add jumps:

- from the old start state to the new start state, and
- from each old final state to the new final state.

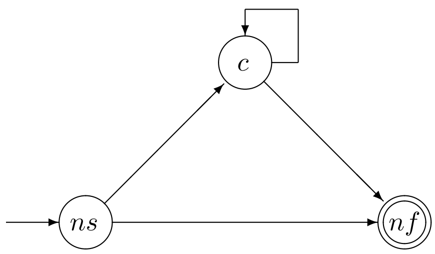
As a result, we get the following automaton:




**Eliminating the error state.** Then, we need to eliminate the intermediate states one by one. Let us start with eliminating the error state $e$. As a result, for all other states $i$ and $j$, we get $R'_{i,j} = R_{i,j} \cup (R_{i,e}R^*_{e,e}R_{e,j})$. By definition of a sink state, it has no arrows going from it to any other states. Thus, we always have $R_{e,j} = \emptyset$. Concatenation with the empty set $R^*_{e,j}$ is empty set, so we always have $R_{i,e}R^*_{e,e}R_{e,j} = \emptyset$. Union of any set with the empty set is that same original set, so we have $R'_{i,j} = R_{i,j}$. Thus, we get the following simplified automaton:




**Eliminating the start state.** First, we draw all possible arrows:



Now, to find expressions to place at all these arrows, we will use the general formula

$$R'_{i,j} = R_{i,j} \cup (R_{i,k} R^*_{k,k} R_{k,j}),$$

where $k$ is the state that we are eliminating, i.e., in this case, the state $k = s$.

By applying this formula, and by using simplification formulas described in the lecture, we get the following results:

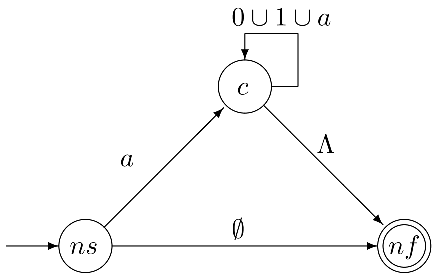
$$R'_{ns,c} = R_{ns,c} \cup (R_{ns,s} R^*_{s,s} R_{s,c}) = \emptyset \cup (\Lambda \emptyset^* a) = \emptyset \cup (\Lambda \Lambda a) = \emptyset \cup a = a;$$

$$R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,s} R^*_{s,s} R_{s,nf}) = \emptyset \cup (\Lambda \emptyset^* \emptyset) = \emptyset \cup \emptyset = \emptyset;$$

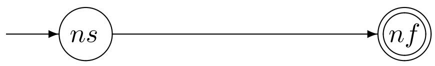
$$R'_{c,c} = R_{c,c} \cup (R_{c,s} R^*_{s,s} R_{s,c}) = 0 \cup 1 \cup a \cup (\emptyset \ldots) = 0 \cup 1 \cup a \cup \emptyset = 0 \cup 1 \cup a;$$

$$R'_{c,nf} = R_{c,nf} \cup (R_{c,s} R^*_{s,s} R_{s,nf}) = \Lambda \cup (\emptyset \ldots) = \Lambda \cup \emptyset = \Lambda.$$

Thus, the 3-state a-automaton takes the following form:




**Eliminating the state $n$.** Now, all that remains to do is to go from here to the 2-state a-automaton by eliminating the remaining state $c$:




The final expression is the corresponding expression for $R'_{ns,nf}$:

$$R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,c} R^*_{c,c} R_{c,nf}) =$$

$$\emptyset \cup (a * 0 \cup 1 \cup a)^* \Lambda) = a(0 \cup 1 \cup a)^* \Lambda = a(0 \cup 1 \cup a)^*.$$

**Resulting answer:** The regular expression corresponding to the original automaton is $a(0 \cup 1 \cup a)^*$.