Task: Apply the general algorithm for transforming the finite automaton into a regular language (i.e., a language described by a regular expression) to the following automaton.

This automaton has two states: $g$ (good student) and $p$ (student on probation): $g$ is the starting state, it is also the final state. The only three symbols are $A$, $B$, and $F$.

- From $g$, $A$ and $B$ lead back to $g$, and $F$ leads to $p$.
- From $p$, any symbol leads back to $p$.

Solution. We start with the described automaton:

```
  A, B  A, B, F
G   F
```

According to the general algorithm, first we add a new start state $ns$ and a few final state $f$, 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.

```
  A, B  A, B, F
ns ε  G   F
```

Then, we need to eliminate the two intermediate states $g$ and $p$ one by one. We can start with eliminating $g$ or with eliminating $p$. Let us show what happens in both cases.

First version, when we first eliminate the state $g$. First, we draw all possible arrows:
Now, to find expressions to place at all these arrows, we will use the general formula

\[ R'_{i,j} = R'_{i,k} \cup (R_{i,k} R_k R_{k,j}), \]

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:

\[
\begin{align*}
R'_{n_s,p} &= R_{n_s,p} \cup (R_{n_s,g} R_{g,p}) = \emptyset \cup (A \cup B)^* F = (A \cup B)^* F; \\
R'_{n_s,n_f} &= R_{n_s,n_f} \cup (R_{n_s,g} R_{g,n_f}) = \emptyset \cup (A \cup B)^* \Lambda = \emptyset; \\
R'_{p,p} &= R_{p,p} \cup (R_{p,g} R_{g,p}) = (A \cup B \cup F) \cup \emptyset = (A \cup B \cup F); \\
R'_{p,n_f} &= R_{p,n_f} \cup (R_{p,g} R_{g,n_f}) = \emptyset \cup \emptyset = \emptyset.
\end{align*}
\]

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

Now, all that remains to do is to go from here to the 2-state a-automaton by eliminating the remaining state \( p \):
The final expression is the corresponding expression for $R'_{ns,nf}$:

$$R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,p}R^*_pR_{p,nf}) = (A \cup B)^* \cup ((A \cup B)^*F(A \cup B \cup F)^*\emptyset) = (A \cup B)^* \cup \emptyset = (A \cup B)^*.$$ 

The formula on the previous line is a regular expression corresponding to the original automaton.

**First version – answer:** $(A \cup B)^*$.

**Second version, when we first eliminate the state p.** First, we draw all possible arrows:

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

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

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

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

- $R'_{ns,g} = R_{ns,g} \cup (R_{ns,p}R^*_pR_{p,g}) = \Lambda \cup \emptyset = \Lambda$;
- $R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,p}R^*_pR_{p,nf}) = \emptyset \cup \emptyset = \emptyset$;
- $R'_{g,g} = R_{g,g} \cup (R_{g,p}R^*_pR_{p,g}) = (A \cup B) \cup (F(A \cup B \cup F)^*\emptyset) = A \cup B \cup \emptyset = A \cup B$;
- $R'_{g,nf} = R_{g,nf} \cup (R_{g,p}R^*_pR_{p,nf}) = \Lambda \cup (F(A \cup B \cup F)^*\emptyset) = \Lambda \cup \emptyset = \Lambda$.

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

$$A \cup B$$
Now, all that remains to do is to go from here to the 2-state a-automaton by eliminating the remaining state \( g \):

![Diagram](image)

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

\[
R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,g}R_{g,g}R_{g,nf}) = \\
\emptyset \cup (\Lambda(A \cup B)^* \Lambda) = (A \cup B)^*.
\]

The formula \((A \cup B)^*\) is also a regular expression corresponding to the original automaton.

**Second version – answer:** \((A \cup B)^*\).