Solution to Homework 1

**Task 1: general description.** In class, we designed automata for recognizing integers and real numbers.

**Task 1.1.** Use the same ideas to describe an automaton for recognizing Java variable names; to describe an automaton, draw a picture like we do in class.

A natural idea is to have 3 states: start ($s$), correct variable name ($v$), and error ($e$). Start is the starting state, $v$ is the only final state. The transitions are as follows:

- from $s$, any letter $a, \ldots, z$ lead to $v$, every other symbol leads to $e$;
- from $v$, any digit, any letter, and the underline symbol lead back to $v$, every other symbol leads to $e$;
- from $e$, every symbol leads back to $e$.

**Solution.** The desired automaton takes the following form:
Task 1.2. Trace, step-by-step, how this finite automaton will check whether the following two words (sequences of symbols) are correct Java variable names:

- the word `size7` (which this automaton should accept) and
- the word `7size` (which this automaton should reject).

Solution. Let us trace how this automaton will accept the word `size7`. We are originally in the state $s$:

Then, we read the first letter $s$ of the word `size7`, so we move to state $v$:

Then, we read the second letter $i$ of the word `size7`, and we stay in the state $v$:  

Then, we read the letter $z$ of the word \textit{size7}, and still stay in $v$:

Same with letter $e$ of the word \textit{size7}:

Finally, we read the digit 7 of the word \textit{size7} and still stay in $v$: 
The word is read, we are in the final state, so the word $size7$ is accepted.

Let us now trace how the automaton will react to the word $7size$. We also start in the start state $s$:

Then, we read the first symbol 7 of the word $7size$, and go to the error state $e$:

After that, we read three other letters and stay in the error state:
We have read all the symbols, we are in the state $e$ which is not final, so the word 7size is not accepted.
Task 1.3. Write down the tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ corresponding to this automaton:

- $Q$ is the set of all the states,
- $\Sigma$ is the alphabet, i.e., the set of all the symbols that this automaton can encounter; for simplicity, consider only five symbols: digits 0, 1, letters $a$ and $b$, and an underscore;
- $\delta : Q \times \Sigma \to Q$ is the function that describes, for each state $q$ and for each symbol $s$, the state $\delta(q, s)$ to which the automaton that was originally in the state $q$ moves when it sees the symbol $s$ (you do not need to describe all possible transitions this way, just describe two of them);
- $q_0$ is the starting state, and
- $F$ is the set of all final states.

Solution. $Q = \{s, v, e\}$, $\Sigma = \{a, b, \ldots, z, 0, 1, \ldots, 9, \_\}$, $q_0 = s$, $F = \{v\}$, and the transition function $\delta$ is described by the following table:

<table>
<thead>
<tr>
<th></th>
<th>$a, b, \ldots, z$</th>
<th>0, 1, $\ldots$, 9, _</th>
<th>$_\ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>$v$</td>
<td>$e$</td>
<td>$e$</td>
</tr>
<tr>
<td>$v$</td>
<td>$v$</td>
<td>$v$</td>
<td>$e$</td>
</tr>
<tr>
<td>$e$</td>
<td>$e$</td>
<td>$e$</td>
<td>$e$</td>
</tr>
</tbody>
</table>
**Task 1.4.** Apply the general algorithm for union and intersection to:

- this automaton as Automaton $A$ and
- to the automaton for recognizing unsigned integers – that is described in the examples lecture – as Automaton $B$.

For simplicity, feel free to assume that you only have symbols $+, 0, 1,$ and $a$.

**Solution.** If we limit ourselves to these 5 symbols, then the Automaton $A$ takes the form

![Automaton A Diagram]

The Automaton $B$ has the following form:

![Automaton B Diagram]
Short solution. In the beginning, before we see any symbols, both automata are in the state $s$, so the combined automaton is in the state $(s, s)$. Then:

- if we read 0 or 1, Automaton $A$ goes into state $e$ and automaton $B$ goes into state $i$, so we go into the state $(e, i)$;
- if we read $a$, then $A$ goes into $v$ and $B$ goes into $e$, so the combined automaton goes into $(v, i)$;
- if we read $+$, then both automata go into $e$ states, so the combined automaton goes into $(e, e)$.

We can similarly describe transitions from these three new states. As a result, we get the following automaton:
**Long solution.** The states for the union and intersection automata have the following form:

\[
\begin{array}{ccc}
(s, s) & (s, t) & (s, e) \\
(v, s) & (v, t) & (v, e) \\
(e, s) & (e, t) & (e, e)
\end{array}
\]

To make the picture clearer, we describe transitions from each state on a separate graph. For transitions from the state \((s, s)\), we get the following picture:
For transitions from the state \((s, i)\), we have the following graph:

Transitions from the state \((s, e)\) have the following form:
Transitions from the state \((v,s)\) have the following form:

\[
\begin{align*}
& (s, s) \\
& (s, i) \\
& (s, e)
\end{align*}
\]

Transitions from the state \((v,i)\) have the following form:

\[
\begin{align*}
& (s, s) \\
& (s, i) \\
& (s, e)
\end{align*}
\]
Transitions from the state \((v,e)\) have the following form:

\[
\begin{align*}
(s,s) & \\
(s,i) & \\
(s,e) & \\
(v,s) & \\
(v,i) & \\
(v,e) & + 0,1,\alpha
\end{align*}
\]

Transitions from the state \((\epsilon,s)\) have the following form:

\[
\begin{align*}
(s,s) & \\
(s,i) & \\
(s,e) & \\
v,s & \\
v,i & \\
v,e & \\
(\epsilon,\epsilon) & 0,1
\end{align*}
\]
Transitions from the state \((e, i)\) have the following form:

\[
\begin{align*}
(s, s) & \quad (s, i) & \quad (s, e) \\
(v, s) & \quad (v, i) & \quad (v, e) \\
(e, s) & \quad (e, i) & \quad (e, e)
\end{align*}
\]

Finally, transitions from the state \((e, e)\) have the following form:

\[
\begin{align*}
(s, s) & \quad (s, i) & \quad (s, e) \\
(v, s) & \quad (v, i) & \quad (v, e) \\
(e, s) & \quad (e, i) & \quad (e, e)
\end{align*}
\]