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 names of Java constants. In Java, this name should start with a capital letter, and contain only capital letters, digits, or the underscore symbol. To describe an automaton, draw a picture like we do in class.

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

- from $s$, any capital letter $A, \ldots, Z$ lead to $v$, every other symbol leads to $e$;
- from $v$, any digit, any capital 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:

![Diagram of automaton for recognizing Java constant names]
**Task 1.2.** Trace, step-by-step, how this finite automaton will check whether the following two words (sequences of symbols) are correct are correct names for Java constants:

- the word \( PI \) (which this automaton should accept) and
- the word \( Pi_2 \) (which this automaton should reject).

**Solution.** Let us trace how this automaton will accept the word \( PI \). We are originally in the state \( s \):

Then, we read the first letter \( P \) of the word \( PI \), so we move to state \( v \):

Then, we read the second letter \( I \) of the word \( PI \), and we stay in the state \( v \):
The word is read, we are in the final state, so the word $PI$ is accepted.
Let us now trace how the automaton will react to the word $Pi_2$. We also start in the start state $s$:

Then, we read the symbol $P$ of the word $Pi_2$, so we move to the state $v$:

After that, we read the symbol $i$ in the word $Pi_2$ and move to state $e$: 

3
Then, we read the underscore symbol and stay in the state $e$:

Finally, we read the last symbol 2 of the word $P_i.2$ and stay in the state $e$:

We have read all the symbols, we are in the state $e$ which is not final, so the word $P_i.2$ is not accepted.
Task 1.3. Write down the tuple \( (Q, \Sigma, \delta, q_0, F) \) 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 four symbols: digit 0, letters a and A, and an underscore;
- \( \delta : Q \times \Sigma \rightarrow Q \) 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 staring state, and
- \( F \) is the set of all final states.

Solution. \( Q = \{s, v, e\} \), \( \Sigma = \{0, a, A, _\} \), \( q_0 = s \), \( F = \{v\} \), and the transition function \( \delta \) is described by the following table:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>a</th>
<th>A</th>
<th>_</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>e</td>
<td>e</td>
<td>v</td>
<td>e</td>
</tr>
<tr>
<td>v</td>
<td>v</td>
<td>e</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>e</td>
<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
- a more realistic version of an automaton for recognizing unsigned integers – that automaton is described in the examples lecture – as Automaton $B$; in the example described in the lecture, we assumed, for simplicity, that, in addition to 0 and 1, only the symbol $a$ is allowed; in reality, any other symbol different from 0 and 1 – including symbols $A$ and underscore – leads to the error state $e$.

For simplicity, in your automaton for recognizing the union and intersection of the two languages, feel free to assume that you only have symbols 0, $a$, $A$, and underscore.

Solution. If we limit ourselves to these 4 symbols, then the Automaton $A$ takes the form

![Automaton A](image)

The Automaton $B$ has the following form:

![Automaton B](image)
**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 $A$, Automaton $A$ goes into state $v$ and automaton $B$ goes into state $e$, so we go into the state $(v, e)$;
- if we read 0, then $A$ goes into $e$ and $B$ goes into $i$, so the combined automaton goes into $(e, i)$;
- if we read $a$ or $\epsilon$, 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:

![Automaton Diagram](attachment:image.png)

Note that no state is final for both automata, so no state is final for the intersection.