10. **[18 total marks]** State Machines and Linear Temporal Logic

(a) Consider the following specification written in Temporal Logic:

\[
\begin{align*}
&\square (\text{Initial} \Rightarrow (\text{Initial} W (\text{WhiteSpace} \lor \text{Letter} \lor \text{Digit} \lor \text{Otherwise}))) \\
&\square ((\text{Initial} \land \text{WhiteSpace}) \Rightarrow \Diamond \text{Initial}) \\
&\square ((\text{Initial} \land \text{Digit}) \Rightarrow \Diamond \text{Num}) \\
&\square ((\text{Initial} \land \text{Letter}) \Rightarrow \Diamond \text{Id}) \\
&\square ((\text{Initial} \land \text{Otherwise}) \Rightarrow \Diamond \text{Error}) \\
&\square (\text{Id} \Rightarrow (\text{Id} W (\text{Letter} \lor \text{Digit} \lor \text{Otherwise}))) \\
&\square ((\text{Id} \land (\text{Letter} \lor \text{Digit})) \Rightarrow \Diamond \text{Id}) \\
&\square ((\text{Id} \land \text{Otherwise}) \Rightarrow \Diamond \text{Initial}) \\
&\square (\text{Num} \Rightarrow (\text{Num} W (\text{Digit} \lor \text{Otherwise}))) \\
&\square ((\text{Num} \land \text{Digit}) \Rightarrow \Diamond \text{Num}) \\
&\square ((\text{Num} \land \text{Otherwise}) \Rightarrow \Diamond \text{Initial}) \\
&\square (\text{Error} \Rightarrow (\text{Error} W (\text{true}))) \\
&\square ((\text{Error} \land \text{true}) \Rightarrow \Diamond \text{Error})
\end{align*}
\]

Draw the specified finite state machine.
(b) Now, recognize that in each state with an *Otherwise* event, *Otherwise* means something different. For any state, *Otherwise* means “any event but the other events that emerge from the same state”. Define each of the three *Otherwisees* in terms of the other predicates.

1. *Otherwise* of Initial:

2. *Otherwise* of Id:

3. *Otherwise* of Num:

(c) In the FSM you made for (a), consider the transition from *Id* to *Initial* under the event *Otherwise*. The basic FSM notation indicates neither any conditions on the transition nor an action to happen when a transition is taken. The UML state machine notation allows specifying both conditions on the transition and an action to happen when a transition is taken.

Assume that *Otherwise(x)* means that the actual otherwise character that triggers the *Otherwise* event is available to be used in the transition’s conditions and actions by mentioning the parameter *x*.

On the transition line in the diagram below, write the UML expression associated with this transition that says

“Whenever in state *Id*, if the input is the otherwise character *x* and the *x* is a punctuation character (*punct(x)*) then first the current value of *token* is emitted (*emit(token)*), and then *token* is assigned the value of *x*. Finally, the next state is *Initial*.”