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

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

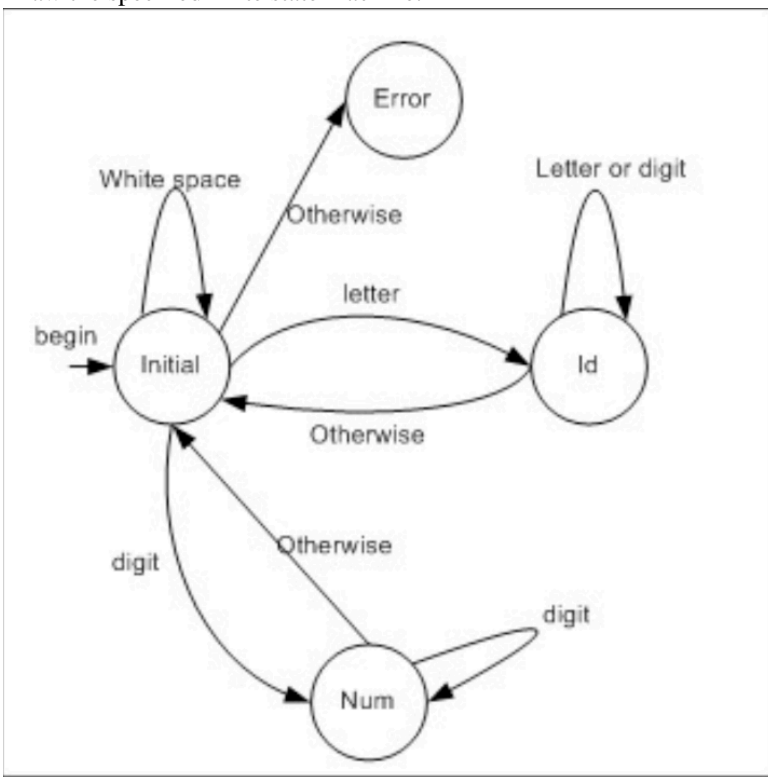
$\Box(Initial \Rightarrow (Initial\ \mathcal{W}\ (WhiteSpace \vee Letter \vee Digit \vee Otherwise)))$
$\Box((Initial \wedge WhiteSpace) \Rightarrow \bigcirc Initial)$
$\Box((Initial \wedge Digit) \Rightarrow \bigcirc Num)$
$\Box((Initial \wedge Letter) \Rightarrow \bigcirc Id)$
$\Box((Initial \wedge Otherwise) \Rightarrow \bigcirc Error)$

$\Box(Id \Rightarrow (Id\ \mathcal{W}\ (Letter \vee Digit \vee Otherwise)))$
$\Box((Id \wedge (Letter \vee Digit)) \Rightarrow \bigcirc Id)$
$\Box((Id \wedge Otherwise) \Rightarrow \bigcirc Initial)$

$\Box(Num \Rightarrow (Num\ \mathcal{W}\ (Digit \vee Otherwise)))$
$\Box((Num \wedge Digit) \Rightarrow \bigcirc Num)$
$\Box((Num \wedge Otherwise) \Rightarrow \bigcirc Initial)$

$\Box(Error \Rightarrow (Error\ \mathcal{W}\ (false)))$
$\Box((Error \wedge true) \Rightarrow \bigcirc Error)$

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 *Otherwise*s in terms of the other predicates.

1. *Otherwise* of *Initial*:
$\neg\ (WhiteSpace \vee Letter \vee Digit)$

2. *Otherwise* of *Id*:
$\neg\ (Letter \vee Digit)$

3. *Otherwise* of *Num*:
$\neg\ (Digit)$

(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*."

$Otherwise(x)[punct(x)]/emit(token);\ token := x$