

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

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

$$\square(\text{Initial} \Rightarrow (\text{Initial } \mathcal{W} (\text{WhiteSpace} \vee \text{Letter} \vee \text{Digit} \vee \text{Otherwise})))$$

$$\square((\text{Initial} \wedge \text{WhiteSpace}) \Rightarrow \bigcirc \text{Initial})$$

$$\square((\text{Initial} \wedge \text{Digit}) \Rightarrow \bigcirc \text{Num})$$

$$\square((\text{Initial} \wedge \text{Letter}) \Rightarrow \bigcirc \text{Id})$$

$$\square((\text{Initial} \wedge \text{Otherwise}) \Rightarrow \bigcirc \text{Error})$$

$$\square(\text{Id} \Rightarrow (\text{Id } \mathcal{W} (\text{Letter} \vee \text{Digit} \vee \text{Otherwise})))$$

$$\square((\text{Id} \wedge (\text{Letter} \vee \text{Digit})) \Rightarrow \bigcirc \text{Id})$$

$$\square((\text{Id} \wedge \text{Otherwise}) \Rightarrow \bigcirc \text{Initial})$$

$$\square(\text{Num} \Rightarrow (\text{Num } \mathcal{W} (\text{Digit} \vee \text{Otherwise})))$$

$$\square((\text{Num} \wedge \text{Digit}) \Rightarrow \bigcirc \text{Num})$$

$$\square((\text{Num} \wedge \text{Otherwise}) \Rightarrow \bigcirc \text{Initial})$$

$$\square(\text{Error} \Rightarrow (\text{Error } \mathcal{W} (\text{true})))$$

$$\square((\text{Error} \wedge \text{true}) \Rightarrow \bigcirc \text{Error})$$

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 *Otherwises* 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*.”

