

# SOEN331: Introduction to Formal Methods for Software Engineering

## Assignment 1 on extended finite state machines

Author's name

April 7, 2021

### 1 Washing Machine formal specification

The EFSM of the washing machine is the tuple  $S = (Q, \Sigma_1, \Sigma_2, q_0, V, \Lambda)$ , where

$Q = \{\text{off}, \text{on}\}$

$\Sigma_1 = \{\text{turn on}, \text{turn off}\}$

$\Sigma_2 = \{\text{beep}, \text{turn light off}\}$

$q_0 : \text{off}$

$\Lambda$ : Transition specifications

1.  $\rightarrow \text{off}$
2.  $\text{off} \xrightarrow{\text{turn on}} \text{on}$
3.  $\text{on} \xrightarrow{\text{turn off} / (\text{beep}; \text{turn light off})} \text{off}$

As *on* is a composite state, it is defined as the tuple  $S = (Q, \Sigma_1, \Sigma_2, q_0, V, \Lambda)$ , where

$Q = \{\text{operating, servicing}\}$

$\Sigma_1 = \{\text{after (10 s), service signal [idle], machine fixed}\}$

$\Sigma_2 = \{\text{blinking, long beep}\}$

$q_0 : \text{operating}$

$\Lambda$ : Transition specifications

1.  $\xrightarrow{\text{after (10 s) / (blinking; long beep)}} \text{operating}$
2.  $\text{operating} \xrightarrow{\text{service signal [idle]}} \text{service}$
3.  $\text{service} \xrightarrow{\text{machine fixed}} \text{operating}$

As *operating* is a composite state, it is defined as the tuple  $S = (Q, \Sigma_1, \Sigma_2, q_0, V, \Lambda)$ , where

$Q = \{\text{idle, standby, active}\}$

$\Sigma_1 = \{\text{light on, start signal or finish button, power off, power on, completion, cancel, cancel [setting]}\}$

$\Sigma_2 = \{\text{turn light on, clear settings, unlock door}\}$

$q_0 : \text{idle}$

$\Lambda$ : Transition specifications

1.  $\xrightarrow{\text{light on / turn light on}} \text{idle}$
2.  $\text{idle} \xrightarrow{\text{start signal or finish button}} \text{active}$
3.  $\text{active} \xrightarrow{\text{cancel}} \text{idle}$
4.  $\text{active} \xrightarrow{\text{completion / unlock door}} \text{idle}$
5.  $\text{active} \xrightarrow{\text{cancel [setting] / clear settings}} \text{idle}$
6.  $\text{active} \xrightarrow{\text{power off}} \text{standby}$
7.  $\text{standby} \xrightarrow{\text{power on}} \text{active}$

The UML state diagram is shown in Figure 1.

As *active* is a composite state, it is defined as the tuple  $S = (Q, \Sigma_1, \Sigma_2, q_0, V, \Lambda)$ , where

$Q = \{\text{setting, washing, rinse, spin}\}$

$\Sigma_1 = \{\text{start-finish, after (3 min), after (2 min)}\}$

$\Sigma_2 = \{\text{lock door}\}$

$q_0 : \text{setting}$

$\Lambda$ : Transition specifications

1.  $\rightarrow \text{setting}$
2.  $\text{setting} \xrightarrow{[\text{door is closed}] \text{ start-finish / lock door}} \text{washing}$
3.  $\text{washing} \rightarrow \text{rinse}$
4.  $\text{rinse} \xrightarrow{\text{after (3 min)}} \text{spin}$
5.  $\text{spin} \xrightarrow{\text{after (2 min)}} \rightarrow$

The EFSM of the *washing* state is the tuple  $S = (Q, \Sigma_1 q_0, V, \Lambda)$ , where

$Q = \{\text{heating, longwash, shortwash}\}$

$\Sigma_1 = \{\text{after (2 min), after (30 min), after (10 min)}\}$

$q_0 : \text{heating}$

$V : \text{currentTemp: } \mathbb{R}.$

$\Lambda$ : Transition specifications

1.  $\rightarrow \text{heating}$
2.  $\text{heating} \xrightarrow{[\text{ct} < \text{desiredTemp}] \text{ after (2 min)}} \text{heating}$
3.  $\text{heating} \xrightarrow{[\text{ct} \geq \text{desiredTemp}] [\text{mode is long}] \text{ after (2 min)}} \text{longwash}$
4.  $\text{heating} \xrightarrow{[\text{ct} \geq \text{desiredTemp}] [\text{mode is short}] \text{ after (2 min)}} \text{shortwash}$
5.  $\text{longwash} \xrightarrow{\text{after (30 min)}} \rightarrow$
5.  $\text{shortwash} \xrightarrow{\text{after (10 min)}} \rightarrow$

The UML state diagram is shown in Figure 2.

## 2 UML state diagrams

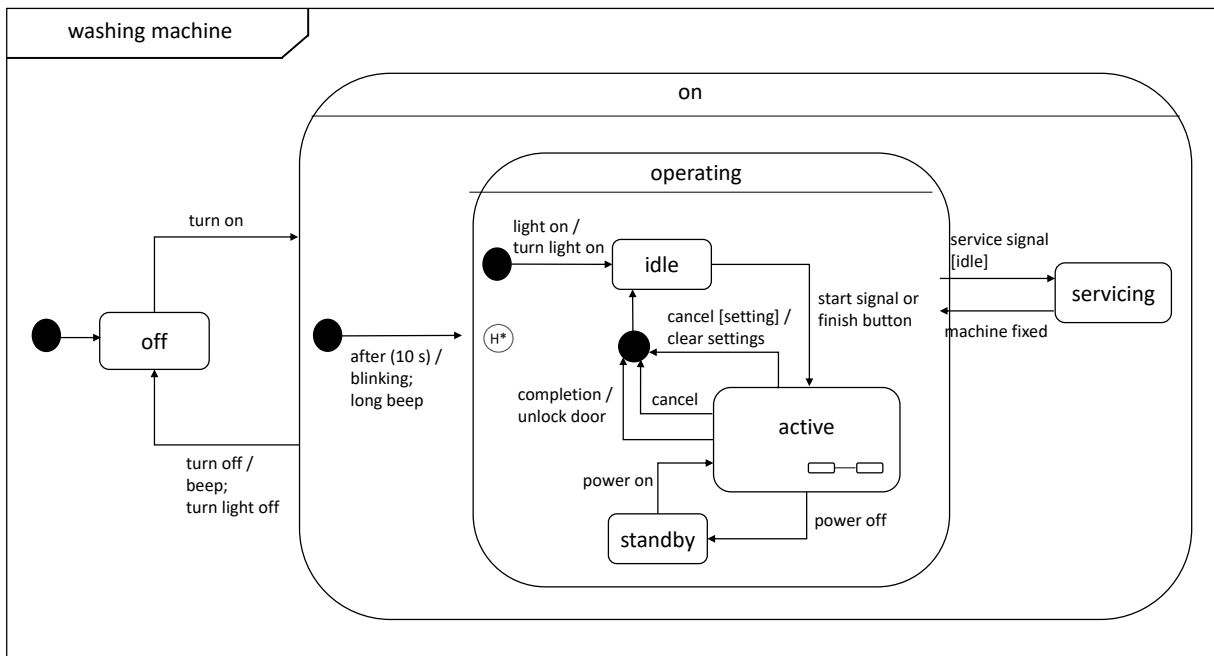


Figure 1: Washing Machine

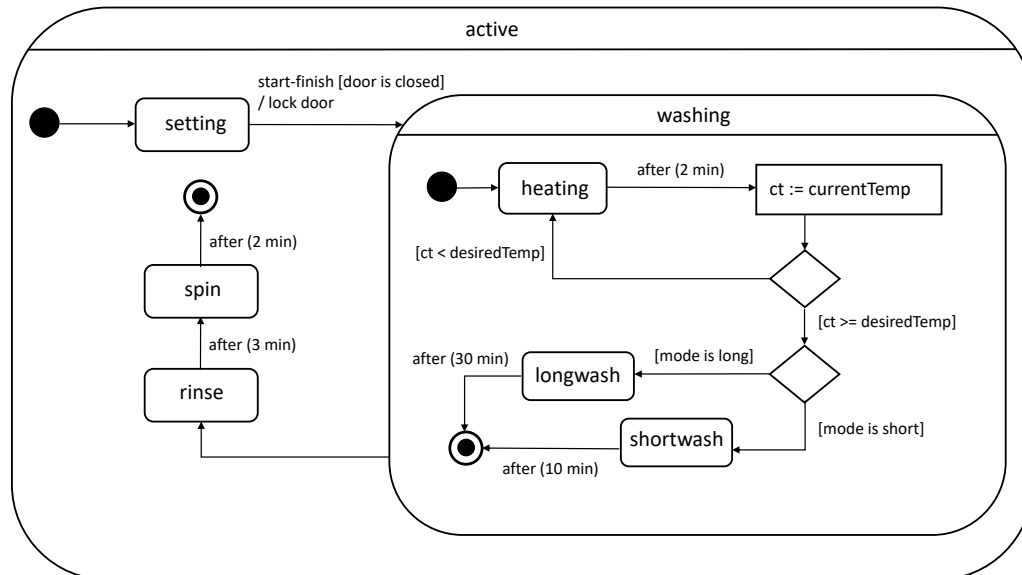


Figure 2: Washing Machine (Active state)