Concordia University

Department of Computer Science and Software Engineering

SOEN 331 - S and U

Introduction to Formal Methods for Software Engineering

Assignment 3 - Solutions

Extended Finite State Machines

Team 19 - Section U

Samuel Boaknin 40009692

Ryan Leyland 40015165

Saleha Tariq 40006997

Meng Susana Ung 40099729

Due Date: April 9, 2021

1 Washing Machine formal specification

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

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

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

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

q_0 : off

Λ: Transition specifications

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

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

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

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

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

q_0 : operating

Λ: Transition specifications

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

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

Λ: Transition specifications

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

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 : setting

V : door = {open, closed}

Λ: Transition specifications

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

As *washing* is a composite state, it is defined as 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 : heating

V : currentTemp, desiredTemperature: $\mathbb{R}$, mode = {short, long}

Λ: Transition specifications

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

The UML state diagram is shown in Figure 2.

2 UML state diagrams

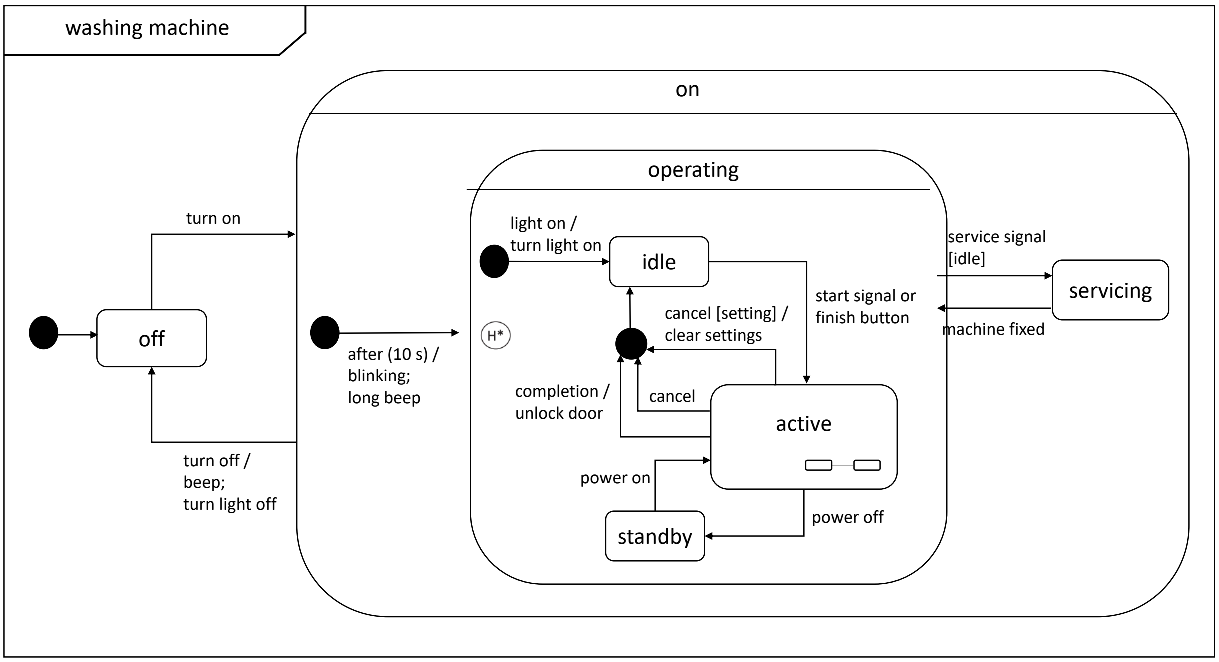

Figure 1: Washing Machine

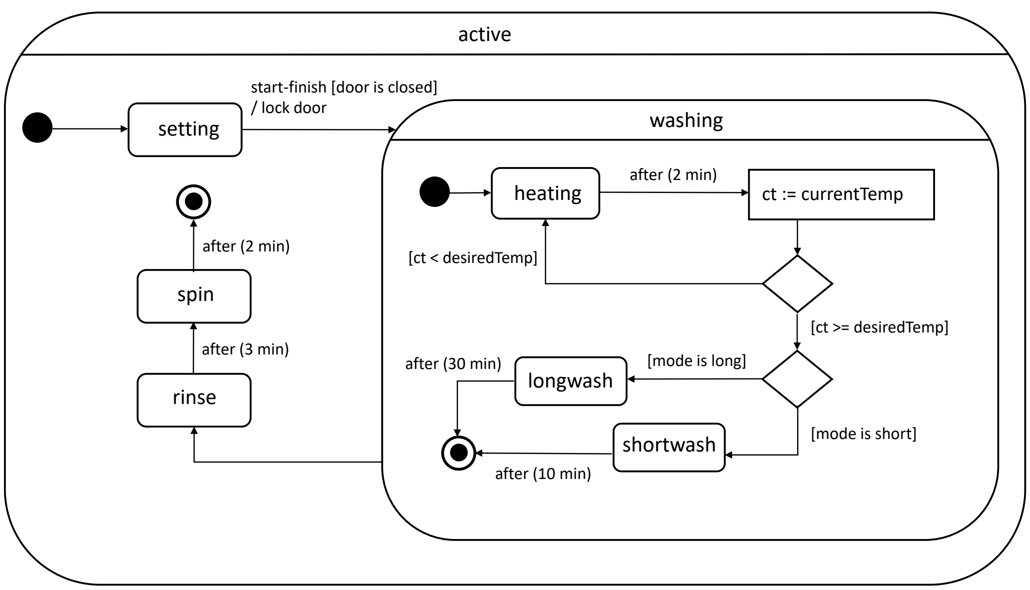

Figure 2: Washing Machine (Active state)