SOEN 331 – Assignment 3

Sam Assaf – 6150748
Jessica Falco – 6597882

We decided to add a top level state called **active** in order to handle the universal kill transition and added it to the specification.

Top-Level

$$S = (Q,\ \Sigma_1,\ \Sigma_2,\ \vee,\ \wedge) \text{ where}$$
$$Q = \{active,\ exit\}$$
$$\Sigma_1 = \{kill\}$$
$$\Sigma_2 = \{\}$$
$$q_0 : active$$
$$\vee = \{\}$$
$$\wedge = \{$$
$$\rightarrow active$$
$$active \xrightarrow{kill} exit$$
$$\}$$

Overall EFSM

$S = (Q,\ \Sigma_1,\ \Sigma_2,\ q_0,\ \vee,\ \wedge)$ where

$Q = \{dormant,\ init,\ idle,\ monitoring,\ error\_diagnosis,\ safe\_shutdown\}$
$\Sigma_1 = \{start,\ init\_ok,\ begin\_monitoring,\ init\_crash,\ retry\_init,\ shutdown,\ sleep,$
$idle\_crash,\ idle\_rescue,\ monitor\_crash,\ moni\_rescue\}$
$\Sigma_2 = \{init\_err\_msg,\ idle\_err\_msg,\ moni\_err\_msg\}$
$q_0 : dormant$
$\vee = \{retry : \mathbb{N}_0\}$
$\wedge = \{$
$\rightarrow dormant$
$dormant \xrightarrow{start/retry=0} init$
$init \xrightarrow{init\_ok/retry=0} idle$
$idle \xrightarrow{begin\_monitoring} monitoring$
$init \xrightarrow{init\_crash/init\_err\_msg} error\_diagnosis$
$error\_diagnosis \xrightarrow{retry\_init[retry<3]/retry++} init$
$idle \xrightarrow{idle\_crash/idle\_err\_msg} error\_diagnosis$
$error\_diagnosis \xrightarrow{idle\_rescue} idle$

$monitoring \xrightarrow{monitor\_crash/moni\_err\_msg} error\_diagnosis$
$error\_diagnosis \xrightarrow{moni\_rescue} monitoring$
$error\_diagnosis \xrightarrow{shutdown[retry\geq 3]} safe\_shutdown$
$safe\_shutdown \xrightarrow{sleep} dormant$
$\}$

Refine init

$S = (Q,\ \Sigma_1,\ \Sigma_2,\ q_0,\ \vee,\ \wedge)$ where

$Q = \{boot\_hw,\ senchk,\ tchk,\ psichk,\ ready\}$
$\Sigma_1 = \{hw\_ok,\ senok,\ t\_ok,\ psi\_ok\}$
$\Sigma_2 = \{\}$
$q_0 : boot\_hw$
$\vee = \{\}$
$\wedge = \{$
$\rightarrow boot\_hw$
$boot\_hw \xrightarrow{hw\_ok} senchk$
$senchk \xrightarrow{senok} tchk$
$tchk \xrightarrow{t\_ok} psichk$
$psichk \xrightarrow{psi\_ok} ready$
$\}$

Refine monitoring

$S = (Q,\ \Sigma_1,\ \Sigma_2,\ q_0,\ \vee,\ \wedge)$ where

$Q = \{monidle,\ regulate\_environment,\ lockdown\}$
$\Sigma_1 = \{no\_contagion,\ after\_100ms,\ contagion\_alert,\ purge\_succ\}$
$\Sigma_2 = \{FACILITY\_CRIT\_MESG\}$
$q_0 : monidle$
$\vee = \{inlockdown : Boolean\}$
$\wedge = \{$
$\rightarrow monidle$
$monidle \xrightarrow{no\_contagion} regulate\_environment$
$regulate\_environment \xrightarrow{after\_100ms} monidle$
$monidle \xrightarrow{contagion\_alert/(FACILITY\_CRIT\_MESG;\ inlockdown=true)} lockdown$
$lockdown \xrightarrow{purge\_succ/inlcokdown=false} monidle$
$\}$

Refine lockdown

$S = (Q,\ \Sigma_1,\ \Sigma_2,\ q_0,\ \vee,\ \wedge)$ where

$Q = \{prep\_vpurge,\ alt\_temp,\ alt\_psi,\ risk\_assess,\ safe\_status,\ exit\}$
$\Sigma_1 = \{initiate\_purge,\ tcyc\_comp,\ psicyc\_comp\}$
$\Sigma_2 = \{lock\_doors,\ unlock\_doors\}$
$q_0 : prep\_vpurge$
$\vee = \{risk : \mathbb{R}_0\}$
$\wedge = \{$
$\rightarrow prep\_vpurge$
$prep\_vpurge \xrightarrow{initiate\_purge/lock\_doors} alt\_temp$
$prep\_vpurge \xrightarrow{initiate\_purge/lock\_doors} alt\_psi$
$alt\_temp \xrightarrow{tcyc\_comp} risk\_assess$
$alt\_psi \xrightarrow{psicyc\_comp} risk\_assess$
$risk\_assess \xrightarrow{[risk \geq 0.01]} prep\_vpurge$
$risk\_assess \xrightarrow{[risk < 0.01]} safe\_status$
$safe\_status \rightarrow exit$
$\}$

Refine error diagnosis

$S = (Q,\ \Sigma_1,\ \Sigma_2,\ q_0,\ \vee,\ \wedge)$ where

$Q = \{error\_rcv,\ applicable\_rescues,\ reset\_module\_data,\ exit\}$
$\Sigma_1 = \{apply\_protocol\_rescues,\ reset\_to\_stable\}$
$\Sigma_2 = \{\}$
$q_0 : error\_rcv$
$\vee = \{err\_protocol\_def : Boolean\}$
$\wedge = \{$
$\rightarrow error\_rcv$
$error\_rcv \xrightarrow{[err\_protocol\_def]} applicable\_rescues$
$error\_rcv \xrightarrow{[!err\_protocol\_def]} reset\_module\_data$
$applicable\_rescues \xrightarrow{apply\_protocol\_rescues} exit$
$reset\_module\_data \xrightarrow{reset\_to\_stable} exit$
$\}$