SOEN331: Introduction to Formal Methods for Software Engineering
Assignment 2 on Extended Finite State Machines

Martin Marcos 40041398,

Samantha Guillemette 26609198,

Deepkumar Patel 40096716

March 3, 2020

1 Driver-less car system formal specification

The EFSM of the driver-less car system is the tuple $S = (Q, \Sigma_1, \Sigma_2, q_0, V, \Lambda)$, where

$Q = \{idle, parked\ mode, manual\ mode, cruise\ mode, panic\ mode, exit\}$

$\Sigma_1 = \{start\ car, cruise\ signal, drive\ signal, switch, arrived, unforseen, panic\ on, parked\ mode\ signal, pan$

$\Sigma_2 = \{system\ start, engine\ idle, beep, system\ off, stop\ car, hazard\ signals\ on, hazard\ signals\ off\}$

$q_0 : idle$

$V : nav\ system : \{set, not\ set, engine\ idle, car\ stopped\}$

Λ: Transition specifications

1. $\rightarrow idle$
2. $idle \xrightarrow{\text{start/system start; engine idle}} parked\ mode$
3. $parked\ mode \xrightarrow{\text{engine off/system off}} exit$
4. $parked\ mode \xrightarrow{\text{cruise signal[not set]/beep}} manual\ mode$
5. $parked\ mode \xrightarrow{\text{cruise signal[set]/beep}} cruise\ mode$
6. $parked\ mode \xrightarrow{\text{drive signal[engine idle]}} manual\ mode$
7. $manual\ mode \xrightarrow{\text{switch[set]}} cruise\ mode$
8. $cruise\ mode \xrightarrow{\text{switch}} manual\ mode$
9. $cruise\ mode \xrightarrow{\text{arrived}} parked\ mode$
9. $cruise\ mode \xrightarrow{\text{unforseen/stop car; hazard signals on}} panic\ mode$
10. $manual\ mode \xrightarrow{\text{parked mode signal[car stopped]}} parked\ mode$
11. $panic\ mode \xrightarrow{\text{panic off/hazard signals off}} parked\ mode$
12. $manual\ mode \xrightarrow{\text{panic on/stop car; hazard signals on}} panic\ mode$

The UML state diagram is shown in Figure 1

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

$Q = \{running, fast, slower, break\ mode, parked\ mode, panic\ mode\}$

$\Sigma_1 = \{accelerate, decelerate, break, parked\ mode\ signal, panic\ on, panic\ off\}$

$\Sigma_2 = \{increase\ speed, decrease\ speed, 0{-}speed, stop\,car, hazard\ signal\ on, hazard\ signal\ off\}$

$q_0 : running$

Λ: Transition specifications

1. $\rightarrow running$
2. $running \xrightarrow{\text{accelerate/increase speed}} faster$
3. $running \xrightarrow{\text{decelerate/decrease speed}} slower$
4. $running \xrightarrow{\text{break/0-speed}} break\ mode$
5. $break\ mode \xrightarrow{\text{accelerate/increase speed}} running$
6. $faster \xrightarrow{\text{decelerate/decrease speed}} slower$
7. $slower \xrightarrow{\text{decelerate/decrease speed}} slower$
8. $slower \xrightarrow{\text{decelerate/decrease speed}} break\ mode$
9. $break\ mode \xrightarrow{\text{parked mode signal}} parked\ mode$
10. $running \xrightarrow{\text{panic on/stop car; hazard signal on}} panic\ mode$
11. $panic\ mode \xrightarrow{\text{panic off/hazard signal off}} parked\ mode$

The UML state diagram is shown in Figure 2

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

$Q = \{tailing\ mode, changing\ lane\ mode, navigation\ mode\}$

$\Sigma_1 = \{i\ to\ c, t\ to\ c, c\ to\ t, c\ to\ n, n\ to\ c\}$

$q_0 : tailing\ mode$

Λ: Transition specifications

1. $\rightarrow tailing\ mode$
2. $\xrightarrow{\text{i to c}} changing\ lane\ mode$
3. $tailing\ mode \xrightarrow{\text{t to c}} changing\ lane\ mode$
4. $changing\ lane\ mode \xrightarrow{\text{c to t}} tailing\ mode$
5. $changing\ lane\ mode \xrightarrow{\text{c to n}} navigation\ mode$
6. $navigation\ mode \xrightarrow{\text{n to c}} changing\ lane\ mode$

The UML state diagram is shown in Figure 3

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

$Q = \{tailing\ start, accelerate, decelerate, changing\ lane\ mode\}$
$\Sigma_1 = \{obstacle\}$
$\Sigma_2 = \{t\ to\ c, maintain\ speed, switch\ lane\}$
$q_0 : tailing\ start$
$V : speedOfCar : \{minSpeedRange, maxSpeedRange\},$
$\quad distanceOfCar, minSpeedRange, maxSpeedRange, minDistance : \mathbb{R}$
Λ: Transition specifications

1. $\rightarrow tailing\ start$
2. $tailing\ start \xrightarrow{\text{[s<minSpeedRange]}} accelerate$
3. $tailing\ start \xrightarrow{\text{[maxSpeedRange <s>minSpeedRange]/t to c; maintain speed}} changing\ lane\ mode$
4. $tailing\ start \xrightarrow{\text{obstacle[d<minDistance]/t to c; switch lane}} changing\ lane\ mode$
5. $tailing\ start \xrightarrow{\text{[s>minSpeedRange or d<minDistance]}} decelerate$

The UML state diagram is shown in Figure 4

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

$Q = \{lane\ start, maintain\ car\ speed, tailing\ mode, change\ lane\ mode, cruise\ mode, panic\ mode\}$
$\Sigma_1 = \{maintain\ speed, switch\ lane, unforseen\}$
$\Sigma_2 = \{c\ to\ t, stop\ car, hazard\ signal\ on\}$
$q_0 : lane\ start$
$V : targetLane : \{car\ in\ t, car\ not\ in\ t\},$
$\quad speedOfCar : \{minSpeedRange, maxSpeedRange\},$

$distanceOfCar, minSpeedRange, maxSpeedRange, minDistance : \mathbb{R}$

Λ: Transition specifications

1. $\rightarrow$ *lane start*
2. $lane\ start \xrightarrow{\text{maintain speed [d>= minDistance]}} maintain\ car\ speed$
3. $maintain\ car\ speed \xrightarrow{\text{[d>= minDistance]}} maintain\ car\ speed$
4. $maintain\ car\ speed \xrightarrow{\text{[d<minDistance]/c to t}} tailing\ mode$
5. $lane\ start \xrightarrow{\text{maintain speed [d<minDistance]/c to t}} tailing\ mode$
6. $lane\ start \xrightarrow{\text{maintain speed [s>maxSpeedRange \& s<minSpeedRange]/c to t}} tailing\ mode$
7. $lane\ start \xrightarrow{\text{switch lane[car not in t]}} change\ lane\ mode$
8. $lane\ start \xrightarrow{\text{switch lane[car in t]/c to n}} cruise\ mode$
9. $lane\ start \xrightarrow{\text{switch lane; unforseen/stop car; hazard signal on}} panic\ mode$
10. $change\ lane\ mode \xrightarrow{\text{[car not in t]}} change\ lane\ mode$
11. $change\ lane\ mode \xrightarrow{\text{[car in t]/c to n}} cruise\ mode$
12. $change\ lane\ mode \xrightarrow{\text{unforseen/stop car; hazard signal on}} panic\ mode$

The UML state diagram is shown in Figure 5

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

$Q = \{navigation\ start, turn\ left, turn\ right, turn\ left\ ahead, turn\ right\ ahead, changing\ lane\ mode, desti$

$\Sigma_1 = \{d\ on\ left, d\ on\ right, TLA, TRA, d\ ahead, car\ at\ d\}$

$\Sigma_2 = \{turn\ left, turn\ right, dest\ ahead, car\ in\ t, n\ to\ c, switch\ lane, arrived\}$

$q_0 : navigation\ start$

$V : targetLane : \{car\ in\ t, car\ not\ in\ t\},$

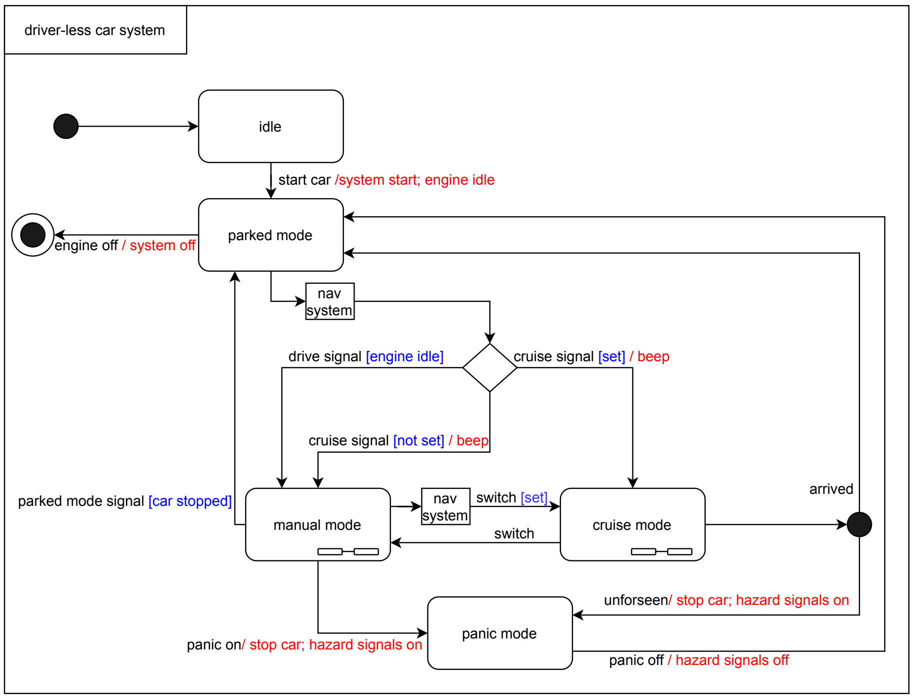
Λ: Transition specifications

1. $\rightarrow$ *mavigation start*
2. $navigation\ start \xrightarrow{\text{d on left/turn left}} turn\ left$
3. $navigation\ start \xrightarrow{\text{d on right/turn right}} turn\ right$
4. $navigation\ start \xrightarrow{\text{TLA[car not in t]}} turn\ left\ ahead$

5. $navigation\ start \xrightarrow{\text{TRA[car not in t]}} turn\ right\ ahead$
6. $navigation\ start \xrightarrow{\text{d ahead/dest ahead; car in t}} turn\ left\ ahead$
7. $turn\ left\ ahead \xrightarrow{\text{/n to c; switch lane}} changing\ lane\ mode$
8. $turn\ right\ ahead \xrightarrow{\text{/n to c; switch lane}} changing\ lane\ mode$
9. $destination\ ahead \xrightarrow{\text{car at d/arrived}} arrived\ destination$

The UML state diagram is shown in Figure 6

2 UML state diagrams



assuming engine idle != car stopped. because while the car is parked and on,
I can still press the gas pedal and make the engine run while the car is still stop/immobile

assuming while having unforseen event in cruise mode,
the car does not immediately stop/hit the breaks that might cause an accident, but gradually stops

Figure 1: Main System.

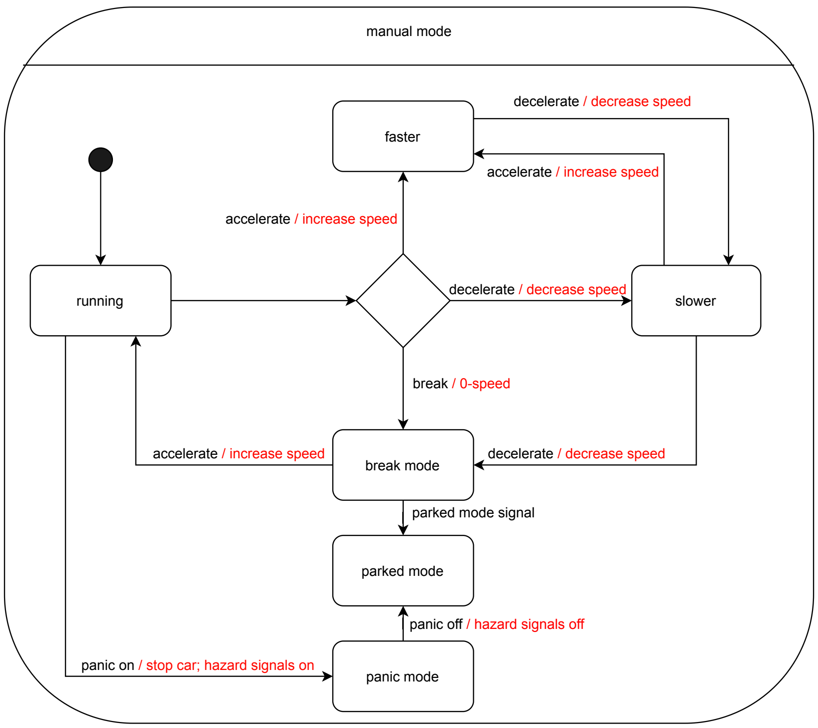


Figure 2: Manual Mode.

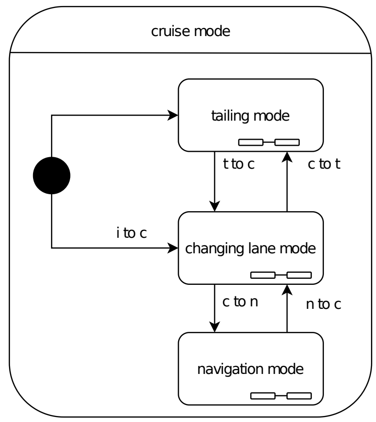


Figure 3: Cruise Mode.

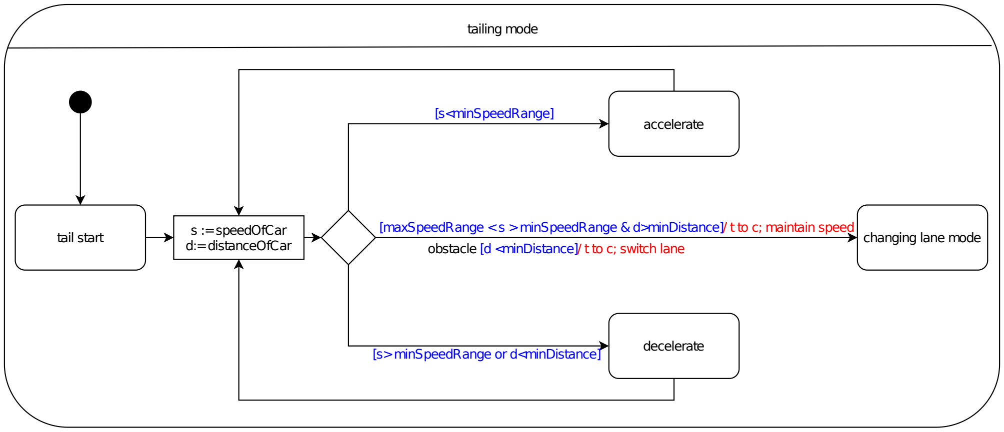


Figure 4: Tailing Mode.

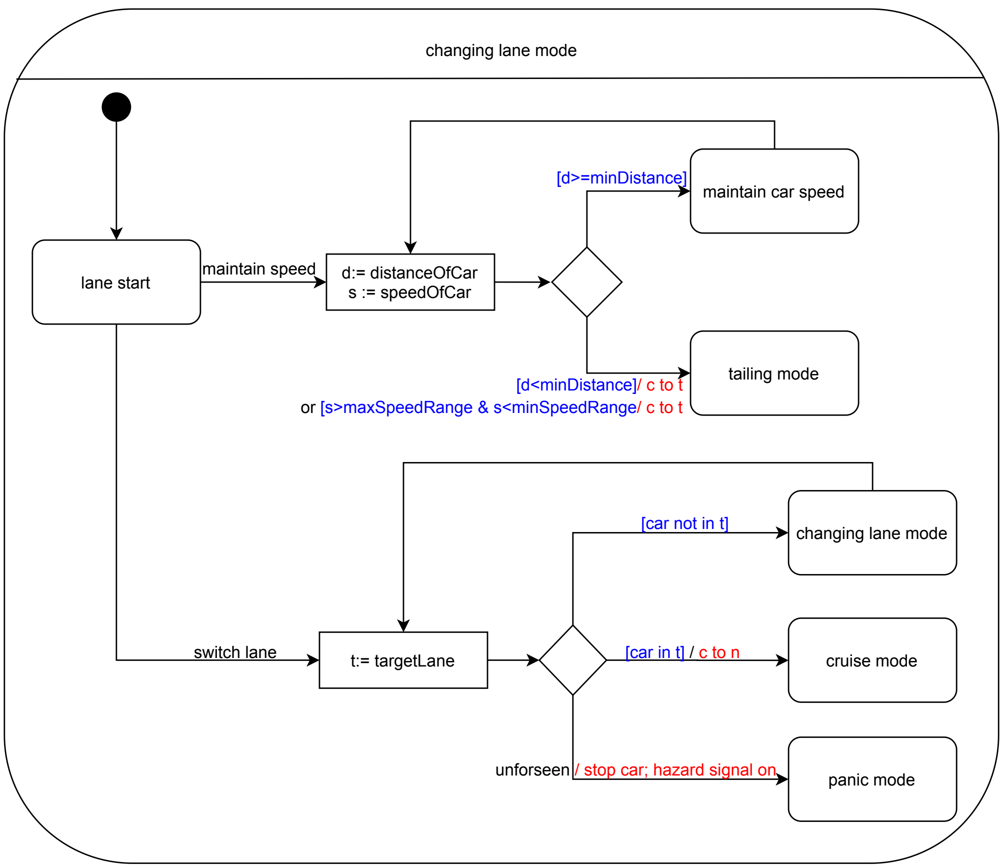


Figure 5: Changing Lane Mode.

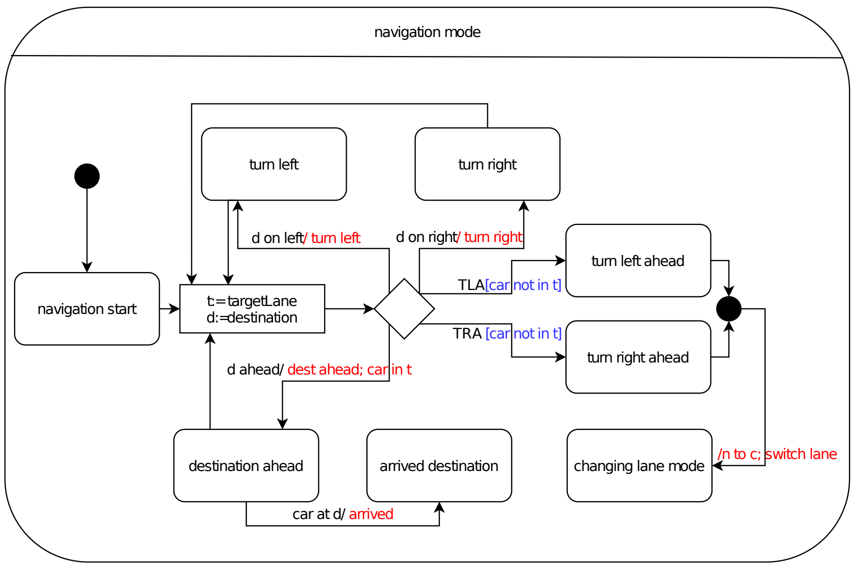


Figure 6: Navigation Mode.