Contents

CONTEXT CruiseCtx

CONSTANTS

CHANGE_BOOL_STATE procedura pentru alternarea unei valori bool

CRUISE_MAX_SPEED viteza maxima admisa a sistemului de croaziera

CRUISE_MIN_SPEED viteza minima admisa a sistemului de croaziera

AXIOMS

axm1: $CHANGE\_BOOL\_STATE \in BOOL \rightarrowtail\!\!\!\!\!\!\rightarrow BOOL$

axm2: $CHANGE\_BOOL\_STATE = \{TRUE \mapsto FALSE, FALSE \mapsto TRUE\}$

axm3: $CRUISE\_MAX\_SPEED \in \mathbb{N}$

axm4: $CRUISE\_MAX\_SPEED = 180000$

valoarea maxima a vitezei de croaziera ce poate fi setata este de 180km/h

axm5: $CRUISE\_MIN\_SPEED \in \mathbb{N}$

axm6: $CRUISE\_MIN\_SPEED = 10000$

valoarea minima a vitezei de croaziera ce poate fi setata este de 10km/h

END

CONTEXT CruiseCtx1
EXTENDS CruiseCtx
CONSTANTS

PEDAL_COMMANDS comenzi posibile pentru apasarea pedalelor

SAFETY_DISTANCE

AXIOMS

axm1: $PEDAL\_COMMANDS = 0 \mathinner{..} 3$

conventie: 0 - nicio comanda; 1 - crestere nivel apasare; 2 - scadere nivel apasare; 3 - scadere forta de apasare la 0

axm2: $SAFETY\_DISTANCE \in \mathbb{N}$

axm3: $SAFETY\_DISTANCE = 50$

distanta de siguranta are valoarea prestabilita de 50 de metri

END

CONTEXT CruiseCtx2
EXTENDS CruiseCtx1
CONSTANTS
VEHICLE_MAX_SPEED
AXIOMS
axm1: $VEHICLE\_MAX\_SPEED \in \mathbb{N}$
axm2: $VEHICLE\_MAX\_SPEED = 200000$
prin constructie vehiculul nostru are o viteza maxima de 200km/h
END

MACHINE CruiseMc
SEES CruiseCtx
VARIABLES
cruise_system_state
cruise_speed
engine_state
INVARIANTS
inv1: $cruise\_system\_state \in BOOL$
inv2: $cruise\_speed \in CRUISE\_MIN\_SPEED \mathinner{..} CRUISE\_MAX\_SPEED$
inv3: $engine\_state \in BOOL$
EVENTS
Initialisation
begin
act1: $cruise\_system\_state := FALSE$
initial sistemul de croaziera este oprit
act2: $cruise\_speed := 50000$
viteza de croaziera prestabilita este de 50km/h
act3: $engine\_state := FALSE$
motorul este oprit
end
Event start_engine ⟨ordinary⟩ $\widehat{=}$
porneste motorul daca este oprit
when
grd1: $engine\_state = FALSE$
then
act1: $engine\_state := CHANGE\_BOOL\_STATE(engine\_state)$
end
Event stop_engine ⟨ordinary⟩ $\widehat{=}$
opreste motorul si sistemul de croaziera daca motorul este pornit
when
grd2: $engine\_state = TRUE$
then
act2: $cruise\_system\_state := FALSE$
act3: $engine\_state := FALSE$
end
Event input_increase_cruise_speed ⟨ordinary⟩ $\widehat{=}$
when
grd1: $engine\_state = TRUE$
motorul este pornit
grd2: $cruise\_system\_state = FALSE$
sistemul de croaziera este oprit
grd3: $cruise\_speed + 2500 \leq CRUISE\_MAX\_SPEED$
verificam daca cresterea vitezei de croaziera depaseste viteza maxima
then
act1: $cruise\_speed := cruise\_speed + 2500$
se adauga valoarea 2.5km/h la viteza de croaziera
end
Event input_decrease_cruise_speed ⟨ordinary⟩ $\widehat{=}$
when
grd1: $engine\_state = TRUE$
motorul este pornit
grd2: $cruise\_system\_state = FALSE$
sistemul de croaziera este oprit
grd3: $cruise\_speed - 2500 \geq CRUISE\_MIN\_SPEED$
verificam daca scaderea valorii vitezei scade sub limita minima
then

act1: $cruise\_speed := cruise\_speed - 2500$
reducem viteza de croaziera cu 2.5km/h

end

Event input_change_cruise_state ⟨ordinary⟩ $\widehat{=}$
porneste sistemul de croaziera odata cu pornirea motorului

when
grd1: $engine\_state = TRUE$

then
act1: $cruise\_system\_state := CHANGE\_BOOL\_STATE(cruise\_system\_state)$

end

END

MACHINE CruiseMc1
REFINES CruiseMc
SEES CruiseCtx1
VARIABLES

acc_pedal_command stocheaza semnalul pentru actuatorul de acceleratie

brake_pedal_command stocheaza semnalul pentru actuatorul de franare

engine_state

cruise_system_state

cruise_speed

distance_sensor semnal pentru distanta curenta fata de vehiculul din fata

prev_distance_sensor stocare pentru distanta anterioara fata de vehiculul din fata

vehicle_board_warning

safety_distance_timer

INVARIANTS

inv1: $acc\_pedal\_command \in PEDAL\_COMMANDS$

inv2: $brake\_pedal\_command \in PEDAL\_COMMANDS$

inv3: $distance\_sensor \in -1 \mathinner{..} SAFETY\_DISTANCE$

valoarea de -1 este valoarea returnata de senzorul de distanta atat timp cat nu avem un vehicul in distanta de siguranta

inv4: $vehicle\_board\_warning \in BOOL$

inv5: $prev\_distance\_sensor \in -1 \mathinner{..} SAFETY\_DISTANCE$

inv6: $safety\_distance\_timer \in \mathbb{N}$

EVENTS

Initialisation ⟨extended⟩

begin

act1: $cruise\_system\_state := FALSE$

initial sistemul de croaziera este oprit

act2: $cruise\_speed := 50000$

viteza de croaziera prestabilita este de 50km/h

act3: $engine\_state := FALSE$

motorul este oprit

act7: $acc\_pedal\_command := 0$

nu se transmite niciun semnal actuatorului de accelerare

act8: $brake\_pedal\_command := 0$

nu se transmite niciun semnal actuatorului de franare

act9: $distance\_sensor := -1$

se asuma faptul ca nu este nicio masina in disanta de siguranta

act10: $prev\_distance\_sensor := -1$

act11: $vehicle\_board\_warning := FALSE$

act12: $safety\_distance\_timer := 0$

end

Event input_change_cruise_state ⟨ordinary⟩ $\widehat{=}$
extends input_change_cruise_state

when

grd1: $engine\_state = TRUE$

then

act1: $cruise\_system\_state := CHANGE\_BOOL\_STATE(cruise\_system\_state)$

end

Event input_decrease_cruise_speed ⟨ordinary⟩ $\widehat{=}$
extends input_decrease_cruise_speed

when

grd1: $engine\_state = TRUE$

motorul este pornit

grd2: $cruise\_system\_state = FALSE$

sistemul de croaziera este oprit

grd3: $cruise\_speed - 2500 \geq CRUISE\_MIN\_SPEED$
verificam daca scaderea valorii vitezei scade sub limita minima

then

act1: $cruise\_speed := cruise\_speed - 2500$
reducem viteza de croaziera cu 2.5km/h

end

Event input_increase_cruise_speed ⟨ordinary⟩ $\widehat{=}$
extends input_increase_cruise_speed

when

grd1: $engine\_state = TRUE$
motorul este pornit
grd2: $cruise\_system\_state = FALSE$
sistemul de croaziera este oprit
grd3: $cruise\_speed + 2500 \leq CRUISE\_MAX\_SPEED$
verificam daca cresterea vitezei de croaziera depaseste viteza maxima

then

act1: $cruise\_speed := cruise\_speed + 2500$
se adauga valoarea 2.5km/h la viteza de croaziera

end

Event start_engine ⟨ordinary⟩ $\widehat{=}$
extends start_engine

when

grd1: $engine\_state = FALSE$

then

act1: $engine\_state := CHANGE\_BOOL\_STATE(engine\_state)$

end

Event stop_engine ⟨ordinary⟩ $\widehat{=}$
extends stop_engine

when

grd2: $engine\_state = TRUE$

then

act2: $cruise\_system\_state := FALSE$
act3: $engine\_state := FALSE$

end

Event execute_increase_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 1$
comanda curenta este de accelerare

then

act2: $acc\_pedal\_command := 0$
reseteaza comanda

end

Event execute_increase_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 1$
comanda curenta pentru actuatorul de franare este apasare

then

act2: $brake\_pedal\_command := 0$
reseteaza comanda

end

Event execute_decrease_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$

grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 2$
comanda curenta pentru actuatorul de acceleratie este scadere apasare

then

act2: $acc\_pedal\_command := 0$
resetare comanda

end

Event execute_decrease_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 2$
comanda curenta pentru actuatorul de franare este apasare

then

act2: $brake\_pedal\_command := 0$
reseteaza comanda

end

Event execute_lift_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 3$
comanda curenta pentru actuatorul de accelerare este eliminare apasare

then

act2: $acc\_pedal\_command := 0$
reseteaza comanda

end

Event execute_lift_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$

when

grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 3$
comanda curenta pentru actuatorul de franare este eliminare apasare

then

act2: $brake\_pedal\_command := 0$
reseteaza comanda

end

Event vehicle_enters_safety_distance ⟨ordinary⟩ $\widehat{=}$

when

grd1: $prev\_distance\_sensor = -1$
la momentul anterior nu exista un vehicul in fata
grd2: $distance\_sensor \neq -1$
o masina a patruns in distanta de siguranta
grd3: $cruise\_system\_state = TRUE$
grd4: $engine\_state = TRUE$

then

act1: $acc\_pedal\_command := 3$
se trimite comanda pentru eliminarea apasarii acceleratiei
act2: $safety\_distance\_timer := 0$
se seteaza cronometrul pentru distanta de siguranta
act3: $prev\_distance\_sensor := distance\_sensor$
se stocheaza distanta curenta ca valoare anterioara
act4: $vehicle\_board\_warning := TRUE$
se trimite un semnal de atentionare soferului

end

Event vehicle_in_safety_distance_before_timer ⟨ordinary⟩ $\widehat{=}$

eveniment declansat cand masina se afla in distanta de siguranta de mai putin de 1 secunda

when

grd1: $distance\_sensor \neq -1$
o masina se afla in fata
grd2: $safety\_distance\_timer < 10$
cronometrul nu a atins o secunda
grd3: $cruise\_system\_state = TRUE$
grd4: $engine\_state = TRUE$
grd5: $prev\_distance\_sensor \neq -1$
vehiculul a fost detectat anterior

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$
se incrementeaza cronometrul cu 0.1 secunde
act2: $prev\_distance\_sensor := distance\_sensor$
actualizam distanta pana la vehicul

end

Event vehicle_in_safety_distance_increased ⟨ordinary⟩ $\widehat{=}$
vehiculul din fata se distanteaza

when

grd1: $distance\_sensor \neq -1$
grd2: $safety\_distance\_timer \geq 10$
vehiculul se afla in distanta de siguranta de mai mult de o secunda
grd3: $cruise\_system\_state = TRUE$
grd4: $engine\_state = TRUE$
grd5: $distance\_sensor > prev\_distance\_sensor$
vehiculul din fata se distanteaza de vehiculul nostru

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$
se incrementeaza cronometrul
act2: $prev\_distance\_sensor := distance\_sensor$
act3: $brake\_pedal\_command := 2$
se trimite comanda de scadere a apasarii franei

end

Event vehicle_in_safety_distance_decreased ⟨ordinary⟩ $\widehat{=}$
vehiculul din fata se apropie

when

grd1: $distance\_sensor \neq -1$
grd2: $safety\_distance\_timer \geq 10$
vehiculul din fata se afla acolo de cel putin o secunda
grd3: $cruise\_system\_state = TRUE$
grd4: $distance\_sensor \leq prev\_distance\_sensor$
vehiculul din fata se apropie de vehiculul nostru
grd5: $engine\_state = TRUE$

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$
se incrementeaza cronometrul
act2: $prev\_distance\_sensor := distance\_sensor$
act3: $brake\_pedal\_command := 1$
se trimite comanda de crestere a franarii

end

Event vehicle_exits_safety_distance ⟨ordinary⟩ $\widehat{=}$
vehiculul din fata iese din detectia senzorului

when

grd1: $distance\_sensor = -1$
grd2: $prev\_distance\_sensor \neq -1$
grd3: $cruise\_system\_state = TRUE$
grd4: $engine\_state = TRUE$

then

act1: $prev\_distance\_sensor := distance\_sensor$
act2: $brake\_pedal\_command := 3$
se trimite comanda de eliminare a apasarii franei

act3: $vehicle\_board\_warning := FALSE$
se sterge avertizarea trimisa soferului

end

Event input_increase_distance_sensor ⟨ordinary⟩ $\widehat{=}$
vehiculul din fata se distanteaza (update valoare senzor)

when
grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor < SAFETY\_DISTANCE$
distanta pana la vehiculul din fata este mai mica decat valoarea prestabilita a distantei de siguranta
grd4: $distance\_sensor \neq -1$

then
act1: $distance\_sensor := distance\_sensor + 1$
creste valoarea detectata cu 1 metru (masina din fata s-a distantat cu 1 metru)

end

Event input_decrease_distance_sensor ⟨ordinary⟩ $\widehat{=}$
vehiculul din fata se apropie (update valoare senzor)

when
grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor > 0$

then
act1: $distance\_sensor := distance\_sensor - 1$
scade valoarea detectata cu 1 metru (masina din fata s-a apropiat cu 1 metru)

end

Event input_exits_range_distance_sensor ⟨ordinary⟩ $\widehat{=}$
resetare senzor cand masina din fata iese din zona de detectie

when
grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = SAFETY\_DISTANCE$
exista o distanta de cel putin SAFETY_DISTANCE metri intre masina noastra si cea din fata

then
act1: $distance\_sensor := -1$
reseteaza senzor pe valoarea "nedetectat"

end

Event cruise_accelerate_vehicle ⟨ordinary⟩ $\widehat{=}$

when
grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = -1$
nu este nicio masina detectata in fata
grd4: $acc\_pedal\_command = 0$

then
act1: $acc\_pedal\_command := 1$
accelereaza vehiculul

end

Event cruise_decelerate_vehicle ⟨ordinary⟩ $\widehat{=}$

when
grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = -1$
nu este nicio masina in fata

then
act1: $acc\_pedal\_command := 2$
scade nivelul de accelerare

end

Event input_enters_range_distance_sensor ⟨ordinary⟩ $\widehat{=}$

when

grd1: $cruise\_system\_state = TRUE$

grd2: $engine\_state = TRUE$

grd3: $distance\_sensor = -1$

then

act1: $distance\_sensor := SAFETY\_DISTANCE$

end

END

MACHINE CruiseMc2
REFINES CruiseMc1
SEES CruiseCtx2
VARIABLES

distance_sensor
prev_distance_sensor
vehicle_board_warning
safety_distance_timer
acc_pedal_command
brake_pedal_command
cruise_speed
cruise_system_state
engine_state
acc_pedal_perc
brake_pedal_perc
vehicle_speed
ONE_PERC_SPEED

INVARIANTS

inv1: $acc\_pedal\_perc \in 0 \,..\, 100$
procentajul apasarii pedalei de acceleratie

inv2: $brake\_pedal\_perc \in 0 \,..\, 100$
procentajul apasarii pedalei de franare

inv3: $vehicle\_speed \in \mathbb{Z}$

inv4: $ONE\_PERC\_SPEED \in \mathbb{N}$

EVENTS

Initialisation ⟨extended⟩

begin

act1: $cruise\_system\_state := FALSE$
initial sistemul de croaziera este oprit

act2: $cruise\_speed := 50000$
viteza de croaziera prestabilita este de 50km/h

act3: $engine\_state := FALSE$
motorul este oprit

act7: $acc\_pedal\_command := 0$
nu se transmite niciun semnal actuatorului de accelerare

act8: $brake\_pedal\_command := 0$
nu se transmite niciun semnal actuatorului de franare

act9: $distance\_sensor := -1$
se asuma faptul ca nu este nicio masina in disanta de siguranta

act10: $prev\_distance\_sensor := -1$

act11: $vehicle\_board\_warning := FALSE$

act12: $safety\_distance\_timer := 0$

act13: $acc\_pedal\_perc := 0$
initial pedala de acceleratie este neapasata

act14: $brake\_pedal\_perc := 0$
initial pedala de franare este neapasata

act15: $vehicle\_speed := 0$
viteza initiala a vehiculului este 0

act16: $ONE\_PERC\_SPEED := VEHICLE\_MAX\_SPEED/100$
aici definim ce inseamna un procent al vitezei masinii

end

Event input_change_cruise_state ⟨ordinary⟩ $\widehat{=}$
extends input_change_cruise_state

when

grd1: $engine\_state = TRUE$

then
act1: $cruise\_system\_state := CHANGE\_BOOL\_STATE(cruise\_system\_state)$
end
Event execute_decrease_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_decrease_acc_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 2$
comanda curenta pentru actuatorul de acceleratie este scadere apasare
grd4: $acc\_pedal\_perc > 0$
then
act2: $acc\_pedal\_command := 0$
resetare comanda
act3: $acc\_pedal\_perc := acc\_pedal\_perc - 1$
se scade procentajul apasarii pedalei de acceleratie cu 1%
end
Event execute_decrease_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_decrease_brake_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 2$
comanda curenta pentru actuatorul de franare este apasare
grd4: $brake\_pedal\_perc > 0$
then
act2: $brake\_pedal\_command := 0$
reseteaza comanda
act3: $brake\_pedal\_perc := brake\_pedal\_perc - 1$
se scade procentajul apasarii pedalei de frana cu 1%
end
Event input_decrease_cruise_speed ⟨ordinary⟩ $\widehat{=}$
extends input_decrease_cruise_speed
when
grd1: $engine\_state = TRUE$
motorul este pornit
grd2: $cruise\_system\_state = FALSE$
sistemul de croaziera este oprit
grd3: $cruise\_speed - 2500 \geq CRUISE\_MIN\_SPEED$
verificam daca scaderea valorii vitezei scade sub limita minima
then
act1: $cruise\_speed := cruise\_speed - 2500$
reducem viteza de croaziera cu 2.5km/h
end
Event execute_increase_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_increase_acc_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 1$
comanda curenta este de accelerare
grd4: $acc\_pedal\_perc < 100$
then
act2: $acc\_pedal\_command := 0$
reseteaza comanda
act3: $acc\_pedal\_perc := acc\_pedal\_perc + 1$
se creste procentajul apasarii pedalei de acceleratie cu 1%
end

Event execute_increase_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_increase_brake_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 1$
comanda curenta pentru actuatorul de franare este apasare
grd4: $brake\_pedal\_perc < 100$
then
act2: $brake\_pedal\_command := 0$
reseteaza comanda
act3: $brake\_pedal\_perc := brake\_pedal\_perc + 1$
se creste procentajul apasarii pedalei de franare cu 1%
end
Event input_increase_cruise_speed ⟨ordinary⟩ $\widehat{=}$
extends input_increase_cruise_speed
when
grd1: $engine\_state = TRUE$
motorul este pornit
grd2: $cruise\_system\_state = FALSE$
sistemul de croaziera este oprit
grd3: $cruise\_speed + 2500 \leq CRUISE\_MAX\_SPEED$
verificam daca cresterea vitezei de croaziera depaseste viteza maxima
then
act1: $cruise\_speed := cruise\_speed + 2500$
se adauga valoarea 2.5km/h la viteza de croaziera
end
Event execute_lift_acc_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_lift_acc_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $acc\_pedal\_command = 3$
comanda curenta pentru actuatorul de accelerare este eliminare apasare
then
act2: $acc\_pedal\_command := 0$
reseteaza comanda
act3: $acc\_pedal\_perc := 0$
end
Event execute_lift_brake_pedal_command ⟨ordinary⟩ $\widehat{=}$
extends execute_lift_brake_pedal_command
when
grd1: $engine\_state = TRUE$
grd2: $cruise\_system\_state = TRUE$
grd3: $brake\_pedal\_command = 3$
comanda curenta pentru actuatorul de franare este eliminare apasare
then
act2: $brake\_pedal\_command := 0$
reseteaza comanda
act3: $brake\_pedal\_perc := 0$
end
Event start_engine ⟨ordinary⟩ $\widehat{=}$
extends start_engine
when
grd1: $engine\_state = FALSE$
then
act1: $engine\_state := CHANGE\_BOOL\_STATE(engine\_state)$
act2: $vehicle\_speed := 1000$

end

Event stop_engine ⟨ordinary⟩ $\widehat{=}$

extends stop_engine

when

grd2: $engine\_state = TRUE$

grd3: $vehicle\_speed < 1000$

then

act2: $cruise\_system\_state := FALSE$

act3: $engine\_state := FALSE$

end

Event vehicle_enters_safety_distance ⟨ordinary⟩ $\widehat{=}$

extends vehicle_enters_safety_distance

when

grd1: $prev\_distance\_sensor = -1$

la momentul anterior nu exista un vehicul in fata

grd2: $distance\_sensor \neq -1$

o masina a patruns in distanta de siguranta

grd3: $cruise\_system\_state = TRUE$

grd4: $engine\_state = TRUE$

then

act1: $acc\_pedal\_command := 3$

se trimite comanda pentru eliminarea apasarii acceleratiei

act2: $safety\_distance\_timer := 0$

se seteaza cronometrul pentru distanta de siguranta

act3: $prev\_distance\_sensor := distance\_sensor$

se stocheaza distanta curenta ca valoare anterioara

act4: $vehicle\_board\_warning := TRUE$

se trimite un semnal de atentionare soferului

end

Event vehicle_in_safety_distance_before_timer ⟨ordinary⟩ $\widehat{=}$

extends vehicle_in_safety_distance_before_timer

when

grd1: $distance\_sensor \neq -1$

o masina se afla in fata

grd2: $safety\_distance\_timer < 10$

cronometrul nu a atins o secunda

grd3: $cruise\_system\_state = TRUE$

grd4: $engine\_state = TRUE$

grd5: $prev\_distance\_sensor \neq -1$

vehiculul a fost detectat anterior

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$

se incrementeaza cronometrul cu 0.1 secunde

act2: $prev\_distance\_sensor := distance\_sensor$

actualizam distanta pana la vehicul

end

Event vehicle_in_safety_distance_increased ⟨ordinary⟩ $\widehat{=}$

extends vehicle_in_safety_distance_increased

when

grd1: $distance\_sensor \neq -1$

grd2: $safety\_distance\_timer \geq 10$

vehiculul se afla in distanta de siguranta de mai mult de o secunda

grd3: $cruise\_system\_state = TRUE$

grd4: $engine\_state = TRUE$

grd5: $distance\_sensor > prev\_distance\_sensor$

vehiculul din fata se distanteaza de vehiculul nostru

grd6: $brake\_pedal\_perc > 100$

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$
se incrementeaza cronometrul
act2: $prev\_distance\_sensor := distance\_sensor$
act3: $brake\_pedal\_command := 2$
se trimite comanda de scadere a apasarii franei

end

Event vehicle_in_safety_distance_decreased ⟨ordinary⟩ $\widehat{=}$
extends vehicle_in_safety_distance_decreased

when

grd1: $distance\_sensor \neq -1$
grd2: $safety\_distance\_timer \geq 10$
vehiculul din fata se afla acolo de cel putin o secunda
grd3: $cruise\_system\_state = TRUE$
grd4: $distance\_sensor \leq prev\_distance\_sensor$
vehiculul din fata se apropie de vehiculul nostru
grd5: $engine\_state = TRUE$
grd6: $brake\_pedal\_perc < 100$

then

act1: $safety\_distance\_timer := safety\_distance\_timer + 1$
se incrementeaza cronometrul
act2: $prev\_distance\_sensor := distance\_sensor$
act3: $brake\_pedal\_command := 1$
se trimite comanda de crestere a franarii

end

Event vehicle_exits_safety_distance ⟨ordinary⟩ $\widehat{=}$
extends vehicle_exits_safety_distance

when

grd1: $distance\_sensor = -1$
grd2: $prev\_distance\_sensor \neq -1$
grd3: $cruise\_system\_state = TRUE$
grd4: $engine\_state = TRUE$

then

act1: $prev\_distance\_sensor := distance\_sensor$
act2: $brake\_pedal\_command := 3$
se trimite comanda de eliminare a apasarii franei
act3: $vehicle\_board\_warning := FALSE$
se sterge avertizarea trimisa soferului

end

Event input_increase_distance_sensor ⟨ordinary⟩ $\widehat{=}$
extends input_increase_distance_sensor

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor < SAFETY\_DISTANCE$
distanta pana la vehiculul din fata este mai mica decat valoarea prestabilita a distantei de siguranta
grd4: $distance\_sensor \neq -1$

then

act1: $distance\_sensor := distance\_sensor + 1$
creste valoarea detectata cu 1 metru (masina din fata s-a distantat cu 1 metru)

end

Event input_decrease_distance_sensor ⟨ordinary⟩ $\widehat{=}$
extends input_decrease_distance_sensor

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor > 0$

then

act1: $distance\_sensor := distance\_sensor - 1$
scade valoarea detectata cu 1 metru (masina din fata s-a apropiat cu 1 metru)

end

Event input_exits_range_distance_sensor ⟨ordinary⟩ $\widehat{=}$

extends input_exits_range_distance_sensor

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = SAFETY\_DISTANCE$
exista o distanta de cel putin SAFETY_DISTANCE metri intre masina noastra si cea din fata

then

act1: $distance\_sensor := -1$
reseteaza senzor pe valoarea "nedetectat"

end

Event cruise_accelerate_vehicle ⟨ordinary⟩ $\widehat{=}$

extends cruise_accelerate_vehicle

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = -1$
nu este nicio masina detectata in fata
grd4: $acc\_pedal\_command = 0$
grd5: $vehicle\_speed < cruise\_speed$
grd6: $brake\_pedal\_perc = 0$
grd7: $acc\_pedal\_perc < 100$

then

act1: $acc\_pedal\_command := 1$
accelereaza vehiculul

end

Event cruise_decelerate_vehicle ⟨ordinary⟩ $\widehat{=}$

extends cruise_decelerate_vehicle

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = -1$
nu este nicio masina in fata
grd4: $vehicle\_speed > cruise\_speed$
grd5: $brake\_pedal\_perc = 0$
grd6: $acc\_pedal\_perc > 0$

then

act1: $acc\_pedal\_command := 2$
scade nivelul de accelerare

end

Event input_enters_range_distance_sensor ⟨ordinary⟩ $\widehat{=}$

extends input_enters_range_distance_sensor

when

grd1: $cruise\_system\_state = TRUE$
grd2: $engine\_state = TRUE$
grd3: $distance\_sensor = -1$

then

act1: $distance\_sensor := SAFETY\_DISTANCE$

end

Event update_speed_with_acc_below ⟨ordinary⟩ $\widehat{=}$

accelereaza vehiculul cand pedala de acceleratie este apasata

when

grd1: $acc\_pedal\_perc > 0$
grd2: $engine\_state = TRUE$

grd3: $vehicle\_speed \leq acc\_pedal\_perc * ONE\_PERC\_SPEED$
vedem daca viteza curenta a vehiculului e mai mica decat trebuie pentru procentul de apasare a acceleratiei

then

act1: $vehicle\_speed := vehicle\_speed + 50 * (acc\_pedal\_perc - (vehicle\_speed * 100 / VEHICLE\_MAX\_SPEED))$

end

Event update_speed_no_acc ⟨ordinary⟩ $\widehat{=}$
decelereaza vehiculul cand acceleratia nu este apasata (pedala de frana poate fi apasata)

when

grd1: $acc\_pedal\_perc = 0$
grd2: $vehicle\_speed > 0$

then

act1: $vehicle\_speed := vehicle\_speed - 50 * (((vehicle\_speed * 100 + ONE\_PERC\_SPEED) / VEHICLE\_MAX\_SPEE$
$brake\_pedal\_perc)$
actualizeaza viteza vehiculului cand se decelereaza

end

Event update_speed_with_acc_above ⟨ordinary⟩ $\widehat{=}$
decelereaza vehiculul atunci cand pedala de acceleratie este apasata dar viteza depaseeste viteza maxima a vehiculului

when

grd1: $acc\_pedal\_perc > 0$
grd2: $engine\_state = TRUE$
grd3: $vehicle\_speed > acc\_pedal\_perc * ONE\_PERC\_SPEED$

then

act1: $vehicle\_speed := vehicle\_speed - 50 * ((vehicle\_speed * 100 / VEHICLE\_MAX\_SPEED) + 1 - acc\_pedal\_perc)$

end

END