Contents

CONTEXT **C0** **2**

CONTEXT **C2** **3**

MACHINE **M0** **4**

MACHINE **M1** **5**

MACHINE **M2** **7**

MACHINE **M3** **11**

MACHINE **M4** **14**

MACHINE **M5** **17**

MACHINE **M6** **21**

CONTEXT C0

SETS

TRAIN

CONSTANTS

a

b

WAY

AXIOMS

axiom1: $\{a, b\} \subseteq \mathbb{N}$

axiom2: $a < b$

axiom3: $WAY = a \mathinner{..} b$

axiom4: $b - a \geq 20$

END

CONTEXT C2
EXTENDS C0
SETS

STATES

CONSTANTS

TTD
VSS
OCCUPIED
FREE
UNKNOW
AMBIGUOUS

AXIOMS

axiom1: $TTD \subseteq \mathbb{P}_1(WAY)$
axiom2: $union(TTD) = WAY$
axiom3: $inter(TTD) = \varnothing$
axiom4: $\forall ttd \cdot (ttd \in TTD \Rightarrow (\exists p, q \cdot (p \,..\, q \subseteq WAY \land p < q \land ttd = p \,..\, q)))$
axiom5: $VSS \subseteq \mathbb{P}_1(WAY)$
axiom6: $union(VSS) = WAY$
axiom7: $inter(VSS) = \varnothing$
axiom8: $\forall vss \cdot (vss \in VSS \Rightarrow (\exists p, q, ttd \cdot (ttd \in TTD \land p \,..\, q \subseteq ttd \land p < q \land vss = p \,..\, q)))$
axiom9: $partition(STATES, \{OCCUPIED\}, \{FREE\}, \{UNKNOW\}, \{AMBIGUOUS\})$

END

MACHINE M0
SEES C0
VARIABLES
  connectedTrain
  front
  rear
INVARIANTS
  inv0_1: $connectedTrain \in TRAIN \rightarrow\!\!\!\!\!\!+ \; BOOL$
  inv0_2: $front \in dom(connectedTrain) \rightarrow WAY$
  inv0_3: $rear \in dom(connectedTrain) \rightarrow\!\!\!\!\!\!+ \; WAY$
  inv0_4: $\forall tr \cdot (tr \in dom(rear) \Rightarrow rear(tr) < front(tr))$
EVENTS
Initialisation
  begin
    act1: $connectedTrain := \varnothing$
    act2: $front := \varnothing$
    act3: $rear := \varnothing$
  end
Event MoveTrainOnTrack ⟨ordinary⟩ $\widehat{=}$
  any
    tr
    len
  where
    grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
    grd2: $len \in \mathbb{N}_1$
    grd3: $front(tr) + len \in WAY$
  then
    act1: $front(tr) := front(tr) + len$
    act2: $rear := (\{TRUE \mapsto rear \Leftarrow\!\!\!\!+ \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
  end
Event _connectTrain ⟨ordinary⟩ $\widehat{=}$
  any
    tr
    fr
    re
    integer
  where
    grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
    grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
    grd2: $fr \in WAY$
    grd3: $integer \in BOOL$
    grd4: $integer = TRUE \Rightarrow re \in WAY$
    grd5: $re < fr$
  then
    act1: $connectedTrain(tr) := TRUE$
    act2: $front(tr) := fr$
    act3: $rear := (\{TRUE \mapsto rear \Leftarrow\!\!\!\!+ \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
  end
Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
  any
    tr
  where
    grd0: $dom(connectedTrain) \neq \varnothing$
    grd1: $tr \in dom(connectedTrain)$
  then
    act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow\!\!\!\!+ \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow\!\!\!\!+ \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$
  end
END

MACHINE M1
REFINES M0
SEES C0
VARIABLES

connectedTrain
front
rear
MA
MAtemp

INVARIANTS

inv1_1: $MA \in dom(connectedTrain) \nrightarrow \mathbb{P}(WAY)$
inv1_2: $\forall tr\cdot(tr \in dom(MA) \Rightarrow (\exists p, q\cdot(p \,..\, q \subseteq WAY \land p \leq q \land MA(tr) = p \,..\, q)))$
inv1_3: $\forall tr\cdot(tr \in dom(MA) \Rightarrow front(tr) \in MA(tr))$
inv1_4: $\forall tr\cdot(tr \in dom(rear) \cap dom(MA) \Rightarrow rear(tr) \in MA(tr))$
inv1_5: $\forall tr1, tr2\cdot((\{tr1, tr2\} \subseteq dom(MA) \land tr1 \neq tr2) \Rightarrow MA(tr1) \cap MA(tr2) = \varnothing)$
inv1_6: $MAtemp \in dom(connectedTrain) \nrightarrow \mathbb{P}(WAY)$
inv1_7: $\forall tr\cdot(tr \in dom(MAtemp) \Rightarrow (\exists p, q\cdot(p \,..\, q \subseteq WAY \land p \leq q \land MAtemp(tr) = p \,..\, q)))$

EVENTS
Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$

end

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

any

tr
p
q

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $p \,..\, q \subseteq WAY \land p \leq q$
grd3: $front(tr) \in p \,..\, q$
grd4: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$
grd5: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MAtemp(tr) := p \,..\, q$

end

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

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MA(tr) := MAtemp(tr)$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$
refines MoveTrainOnTrack

any

tr
len

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$

grd2: $len \in \mathbb{N}_1$

grd3: $front(tr) + len \in MA(tr)$

then

act1: $front(tr) := front(tr) + len$

act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr)+len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

end

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

extends _connectTrain

any

tr

fr

re

$integer$

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$

grd1: $tr \in TRAIN \setminus dom(connectedTrain)$

grd2: $fr \in WAY$

grd3: $integer \in BOOL$

grd4: $integer = TRUE \Rightarrow re \in WAY$

grd5: $re < fr$

then

act1: $connectedTrain(tr) := TRUE$

act2: $front(tr) := fr$

act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$

end

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

extends _toggleTrainConnexionStatus

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$

grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

END

MACHINE M2
REFINES M1
SEES C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS

INVARIANTS

inv2_1: $stateTTD \in TTD \to \{OCCUPIED, FREE\}$
inv2_2: $stateVSS \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
inv2_3: $\forall ttd, tr \cdot ((tr \in dom(front) \setminus dom(rear) \land ttd \in TTD \land front(tr) \in ttd) \Rightarrow stateTTD(ttd) = OCCUPIED)$
inv2_4: $\forall ttd, tr \cdot ((tr \in dom(rear) \land ttd \in TTD \land (rear(tr) \,..\, front(tr)) \cap ttd \neq \varnothing) \Rightarrow stateTTD(ttd) = OCCUPIED)$
inv2_5: $\forall tr1, tr2 \cdot ((tr1 \in dom(rear) \land tr2 \in dom(rear) \land tr1 \neq tr2) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr2) \,..\, front(tr2)) = \varnothing)$
inv2_6: $\forall tr1, tr2 \cdot ((tr1 \in dom(rear) \land tr2 \in dom(front) \setminus dom(rear) \land tr1 \neq tr2) \Rightarrow front(tr2) < rear(tr1))$
inv2_7: $\forall tr1, tr2, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr2 \in dom(front) \setminus dom(rear) \land tr1 \neq tr2 \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr2) \notin ttd)$

EVENTS
Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$

end

Event ComputeTrainMAFollowingTTDStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeTrainMA

any

tr
ttds
p
q
ttds1
p0
p1
q1 ttds1 designe l'ensemble des ttd sur lesquels le train est succeptible de se trouver

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd3: $union(ttds) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $ttds1 \subseteq TTD$
grd6: $union(ttds1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p \,..\, q \subseteq union(ttds \cup ttds1)$

grd10: $p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p\,..\,q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q$

then

act1: $MAtemp(tr) := p\,..\,q$

end

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

refines ComputeTrainMA

any

tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1\,..\,q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0\,..\,(p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p\,..\,q \subseteq union(vsss \cup vsss1)$
grd10: $p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p\,..\,q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q$

then

act1: $MAtemp(tr) := p\,..\,q$
act2: $stateVSS := newstateVSS$

end

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

extends AssignMAtoTrain

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MA(tr) := MAtemp(tr)$

end

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

extends MoveTrainFollowingItsMA

any

tr
len
ttds

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$

grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr) + len \,..\, front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then
act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any
tr
fr
re
integer
ttds

where
grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((fr \in ttd) \lor ((integer = TRUE) \land ((re \,..\, fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (re \,..\, fr) = \varnothing))$
grd9: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < re))$
grd10: $(integer = FALSE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow fr < rear(tr1)))$
grd11: $(integer = FALSE) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any
tr

where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then
act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

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

any

ttd

where

grd0: $ttd \in stateTTD^{-1}[\{OCCUPIED\}]$

grd1: $\forall tr \cdot (tr \in (dom(front) \setminus dom(rear)) \Rightarrow front(tr) \notin ttd)$

grd2: $\forall tr \cdot (tr \in dom(rear) \Rightarrow (rear(tr) \,..\, front(tr)) \cap ttd = \varnothing)$

then

act1: $stateTTD(ttd) := FREE$

end

END

MACHINE M3
REFINES M2
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed

INVARIANTS

inv3_1: $newstateVSScomputed \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$

end

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

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event ComputeTrainMA ⟨ordinary⟩ $\widehat{=}$
refines ComputeTrainMAFollowingVSSStates

any

tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS = newstateVSScomputed$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$

grd9: $p\,..\,q \subseteq union(vsss \cup vsss1)$
grd10: $p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p\,..\,q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q$

then
act1: $MAtemp(tr) := p\,..\,q$
act2: $stateVSS := newstateVSS$

end

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

any
tr

where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then
act1: $MA(tr) := MAtemp(tr)$

end

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

any
tr
len
$ttds$

where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len\,..\,front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1)\,..\,front(tr1)) \cap (rear(tr) + len\,..\,front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then
act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any
tr
fr
re
$integer$
$ttds$

where
grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$

grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee ((integer = TRUE) \wedge ((re \mathrel{..} fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \mathrel{..} front(tr1)) \cap (re \mathrel{..} fr) = \varnothing))$
grd9: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr1) < re))$
grd10: $(integer = FALSE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow fr < rear(tr1)))$
grd11: $(integer = FALSE) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then

act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \lhd\!\!\!- \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \lhd\!\!\!- (ttds \times \{OCCUPIED\})$

end

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

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \lhd\!\!\!- \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \lhd\!\!\!- \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

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

any

ttd

where

grd0: $ttd \in stateTTD^{-1}[\{OCCUPIED\}]$
grd1: $\forall tr \cdot (tr \in (dom(front) \setminus dom(rear)) \Rightarrow front(tr) \notin ttd)$
grd2: $\forall tr \cdot (tr \in dom(rear) \Rightarrow (rear(tr) \mathrel{..} front(tr)) \cap ttd = \varnothing)$

then

act1: $stateTTD(ttd) := FREE$

end

END

MACHINE M4
REFINES M3
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed

EVENTS
Initialisation
begin
act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$
end
Event ComputeVSSStatesFollowingTTDStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStates
any
newstateVSScomputed1
where
grd0: $newstateVSScomputed1 \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
then
act1: $newstateVSScomputed := newstateVSScomputed1$
end
Event ComputeVSSStateswoTTDStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStates
any
newstateVSScomputed1
where
grd0: $newstateVSScomputed1 \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
then
act1: $newstateVSScomputed := newstateVSScomputed1$
end
Event ComputeTrainMA ⟨ordinary⟩ $\widehat{=}$
extends ComputeTrainMA
any
tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver
where
grd0: $newstateVSS = newstateVSScomputed$

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p \,..\, q \subseteq union(vsss \cup vsss1)$
grd10: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p \,..\, q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$

then

act1: $MAtemp(tr) := p \,..\, q$
act2: $stateVSS := newstateVSS$

end

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

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MA(tr) := MAtemp(tr)$

end

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

any

tr
len
$ttds$

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr) + len \in ttd) \vee (tr \in dom(rear) \wedge ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr) + len \,..\, front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \lhd\!\!- \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \lhd\!\!- (ttds \times \{OCCUPIED\})$

end

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

any

tr

fr
re
$integer$
$ttds$

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee ((integer = TRUE) \wedge ((re \,..\, fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (re \,..\, fr) = \varnothing))$
grd9: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr1) < re))$
grd10: $(integer = FALSE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow fr < rear(tr1)))$
grd11: $(integer = FALSE) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then

act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

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

any

ttd

where

grd0: $ttd \in stateTTD^{-1}[\{OCCUPIED\}]$
grd1: $\forall tr \cdot (tr \in (dom(front) \setminus dom(rear)) \Rightarrow front(tr) \notin ttd)$
grd2: $\forall tr \cdot (tr \in dom(rear) \Rightarrow (rear(tr) \,..\, front(tr)) \cap ttd = \varnothing)$

then

act1: $stateTTD(ttd) := FREE$

end

END

MACHINE M5
REFINES M4
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed
newstateVSScomputedTmp

INVARIANTS

inv5_1: $newstateVSScomputedTmp \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$
act9: $newstateVSScomputedTmp := VSS \times \{UNKNOW\}$

end

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

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{UNKNOW\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{OCCUPIED\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{AMBIGUOUS\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{FREE\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

Event updateVSSStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStatesFollowingTTDStates

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 = newstateVSScomputedTmp$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

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

any

tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS = newstateVSScomputed$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$

grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p\,..\,q \subseteq union(vsss \cup vsss1)$
grd10: $p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p\,..\,q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q$

then

act1: $MAtemp(tr) := p\,..\,q$
act2: $stateVSS := newstateVSS$

end

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

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MA(tr) := MAtemp(tr)$

end

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

any

tr
len
$ttds$

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr)+len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr)+len\,..\,front(tr)+len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1)\,..\,front(tr1)) \cap (rear(tr)+len\,..\,front(tr)+len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr)+len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr)+len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr)+len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \lhd\!\!- \{tr \mapsto rear(tr)+len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \lhd\!\!- (ttds \times \{OCCUPIED\})$

end

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

any

tr
fr
re
$integer$
$ttds$

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$

grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee ((integer = TRUE) \wedge ((re \,..\, fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (re \,..\, fr) = \varnothing))$
grd9: $(integer = TRUE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr1) < re))$
grd10: $(integer = FALSE) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow fr < rear(tr1)))$
grd11: $(integer = FALSE) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any
tr

where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then
act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

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

any
ttd

where
grd0: $ttd \in stateTTD^{-1}[\{OCCUPIED\}]$
grd1: $\forall tr \cdot (tr \in (dom(front) \setminus dom(rear)) \Rightarrow front(tr) \notin ttd)$
grd2: $\forall tr \cdot (tr \in dom(rear) \Rightarrow (rear(tr) \,..\, front(tr)) \cap ttd = \varnothing)$

then
act1: $stateTTD(ttd) := FREE$

end

END

MACHINE M6
REFINES M5
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed
newstateVSScomputedTmp
freeVssChangingtoFree
freeVssChangingtoUnknow
freeVssChangingtoOccupied
freeVssChangingtoAmbiguous

INVARIANTS

inv6_1: $freeVssChangingtoFree \subseteq VSS$
inv6_2: $freeVssChangingtoUnknow \subseteq VSS$
inv6_3: $freeVssChangingtoOccupied \subseteq VSS$
inv6_4: $freeVssChangingtoAmbiguous \subseteq VSS$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$
act9: $newstateVSScomputedTmp := VSS \times \{UNKNOW\}$
act10: $freeVssChangingtoFree := \varnothing$
act11: $freeVssChangingtoUnknow := \varnothing$
act12: $freeVssChangingtoOccupied := \varnothing$
act13: $freeVssChangingtoAmbiguous := \varnothing$

end

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

any

vssTtdFree

where

grd1: $vssTtdFree \subseteq stateVSS^{-1}[\{FREE\}]$
grd2: $\forall vss \cdot (vss \in vssTtdFree \Rightarrow vss \subseteq union(stateTTD^{-1}[\{FREE\}]))$

then

act1: $freeVssChangingtoFree := freeVssChangingtoFree \cup vssTtdFree$

end

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

any

vssTtdOccupiedwithNoTrain

where

grd1: $vssTtdOccupiedwithNoTrain \subseteq stateVSS^{-1}[\{FREE\}]$
grd2: $\forall vss \cdot (vss \in vssTtdOccupiedwithNoTrain \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$

grd3: $\forall vss, p, q \cdot ((vss \in vssTtdOccupiedwithNoTrain \land p \,..\, q \in TTD \land vss \subseteq p \,..\, q) \Rightarrow (\forall tr \cdot tr \in connectedTrain^{-1}[\{TRUE\}] \land tr \in dom(rear) \Rightarrow (front(tr) < p \lor rear(tr) > q)))$
grd4: $\forall vss, p, q \cdot ((vss \in vssTtdOccupiedwithNoTrain \land p \,..\, q \in TTD \land vss \subseteq p \,..\, q) \Rightarrow (\forall tr \cdot tr \in connectedTrain^{-1}[\{TRUE\}] \land tr \notin dom(rear) \Rightarrow (front(tr) < p \lor front(tr) > q)))$

then

act1: $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vssTtdOccupiedwithNoTrain$

end

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

any

vssTtdOccupiedwithNoMA

where

grd1: $vssTtdOccupiedwithNoMA \subseteq stateVSS^{-1}[\{FREE\}]$
grd2: $\forall vss \cdot (vss \in vssTtdOccupiedwithNoMA \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$
grd3: $\forall vss, ttd \cdot ((vss \in vssTtdOccupiedwithNoMA \land ttd \in TTD \land vss \subseteq ttd) \Rightarrow (union(ran(MA)) \cap ttd = \varnothing))$

then

act1: $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vssTtdOccupiedwithNoMA$

end

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

refines ComputeStatesOfVSSinFreeState

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{FREE\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$
grd3: $freeVssChangingtoFree \subseteq vss2$
lorsque toutes les transitions seront implementees, ceci deviendra une egalite
grd4: $freeVssChangingtoUnknow \subseteq vss4$
lorsque toutes les transitions seront implementees, ceci deviendra une egalite

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \lhd ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

extends ComputeStatesOfVSSinUnknowState

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{UNKNOW\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \lhd ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

extends ComputeStatesOfVSSinOccupiedState

any

vss

vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{OCCUPIED\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

any

vss
vss1
vss2
vss3
vss4

where

grd1: $vss = stateVSS^{-1}[\{AMBIGUOUS\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$

then

act1: $newstateVSScomputedTmp := newstateVSScomputedTmp \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

end

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

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 = newstateVSScomputedTmp$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

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

any

tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS = newstateVSScomputed$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p \,..\, q \subseteq union(vsss \cup vsss1)$

grd10: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p \,..\, q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$

then

act1: $MAtemp(tr) := p \,..\, q$
act2: $stateVSS := newstateVSS$

end

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

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$

then

act1: $MA(tr) := MAtemp(tr)$

end

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

any

tr
len
ttds

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr) + len \,..\, front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

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

any

tr
fr
re
integer
ttds

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$

grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \land ((fr \in ttd) \lor ((integer = TRUE) \land ((re\,..\,fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $(integer = TRUE) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1)\,..\,front(tr1)) \cap (re\,..\,fr) = \varnothing))$
grd9: $(integer = TRUE) \Rightarrow (\forall tr1\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < re))$
grd10: $(integer = FALSE) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow fr < rear(tr1)))$
grd11: $(integer = FALSE) \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \mathbin{\lhd\!\!-} \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \mathbin{\lhd\!\!-} (ttds \times \{OCCUPIED\})$

end

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

any
tr

where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then
act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \mathbin{\lhd\!\!-} \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \mathbin{\lhd\!\!-} \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

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

any
ttd

where
grd0: $ttd \in stateTTD^{-1}[\{OCCUPIED\}]$
grd1: $\forall tr\cdot(tr \in (dom(front) \setminus dom(rear)) \Rightarrow front(tr) \notin ttd)$
grd2: $\forall tr\cdot(tr \in dom(rear) \Rightarrow (rear(tr)\,..\,front(tr)) \cap ttd = \varnothing)$

then
act1: $stateTTD(ttd) := FREE$

end

END