Contents

CONTEXT c_sd_bolus

SETS

SD

SDF

CONSTANTS

deliver

off

suspend

preempt

s

d

AXIOMS

axm1: $partition(SD, \{deliver\}, \{off\}, \{suspend\}, \{preempt\})$

axm2: $partition(SDF, \{s\}, \{d\})$

END

CONTEXT c_bolus
SETS
BOLUS_STATUS
CONSTANTS
null
normal
square
dual
AXIOMS
axm1: $partition(BOLUS\_STATUS, \{null\}, \{normal\}, \{square\}, \{dual\})$
END

CONTEXT c_normalbolus
CONSTANTS
normal_bolus_rate
AXIOMS
axm1: $normal\_bolus\_rate > 0$
END

CONTEXT c_normalbolus_anim
EXTENDS c_normalbolus
AXIOMS
axm1: $normal\_bolus\_rate = 2$
END

CONTEXT c_basal

SETS

BASALMODE

CONSTANTS

basal_max

c

suspended

delivering

stop

AXIOMS

axm1: $basal\_max \in \mathbb{N}_1$

axm2: $c = 48$

axm3: $partition(BASALMODE, \{suspended\}, \{delivering\}, \{stop\})$

END

CONTEXT c_basal2
EXTENDS c_basal
SETS
PROG0
CONSTANTS
null
call_get_min
return_get_min
call_get_max
return_get_max
PROC_BASAL
AXIOMS
axm2: $PROC\_BASAL \subseteq PROG0$
axm1: $partition(PROC\_BASAL, \{null\}, \{call\_get\_min\}, \{return\_get\_min\}, \{call\_get\_max\}, \{return\_get\_max\})$

END

CONTEXT c_basal2_USER_anim
EXTENDS c_basal2
CONSTANTS
bsuspend
bresume
AXIOMS
axm1: $bsuspend = 3$
axm2: $bresume = 8$
END

MACHINE Square
SEES c_sd_bolus
VARIABLES
state
INVARIANTS
inv1: $state \in SD$
EVENTS
Initialisation
begin
act1: $state := off$
end
Event start ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = off$
then
act1: $state := deliver$
end
Event finish ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = deliver$
then
act1: $state := off$
end
Event suspend ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = deliver$
then
act1: $state := suspend$
end
Event resume ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = suspend$
then
act1: $state := off$
end
Event preempted ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = deliver$
then
act1: $state := preempt$
end
Event resume_from_preempt ⟨ordinary⟩ $\widehat{=}$
when
grd1: $state = preempt$
then
act1: $state := deliver$
end
END

MACHINE Square1
REFINES Square
SEES c_sd_bolus
VARIABLES

state
s_r
s_t
s_deliver_time
s_deliver_rate

INVARIANTS

inv1: $s\_r \in \mathbb{N}$
inv2: $s\_t \in \mathbb{N}$
inv3: $s\_deliver\_time \in \mathbb{N}$
inv4: $s\_deliver\_rate \in \mathbb{N}$
inv5: $state = off \lor state = suspend \Rightarrow s\_deliver\_time = 0 \land s\_deliver\_rate = 0$
inv6: $state = deliver \Rightarrow s\_deliver\_time \geq 0 \land s\_deliver\_rate > 0$
inv7: $state = preempt \Rightarrow s\_deliver\_time \geq 0 \land s\_deliver\_rate = 0$
inv8: $state = off \lor state = suspend \Rightarrow s\_r = 0 \land s\_t = 0$
inv9: $state = deliver \lor state = preempt \Rightarrow s\_r > 0$
inv10: $state = deliver \lor state = preempt \Rightarrow s\_t > 0$

EVENTS
Initialisation ⟨extended⟩

begin
act1: $state := off$
act2: $s\_r := 0$
act3: $s\_t := 0$
act4: $s\_deliver\_time := 0$
act5: $s\_deliver\_rate := 0$
end

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

any
t
r
where
grd1: $state = off$
grd2: $t \in \mathbb{N}_1$
grd3: $r \in \mathbb{N}_1$
then
act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act4: $s\_deliver\_time := t$
act5: $s\_deliver\_rate := r$
end

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

when
grd1: $state = deliver$
then
act1: $state := off$
act2: $s\_deliver\_time := 0$
act3: $s\_deliver\_rate := 0$
act4: $s\_r := 0$
act5: $s\_t := 0$

end
Event suspend ⟨ordinary⟩ $\widehat{=}$
extends suspend
when
grd1: $state = deliver$
then
act1: $state := suspend$
act2: $s\_deliver\_time := 0$
act3: $s\_deliver\_rate := 0$
act4: $s\_r := 0$
act5: $s\_t := 0$
end
Event resume ⟨ordinary⟩ $\widehat{=}$
extends resume
when
grd1: $state = suspend$
then
act1: $state := off$
end
Event preempted ⟨ordinary⟩ $\widehat{=}$
extends preempted
any
t time left for square bolus
where
grd1: $state = deliver$
grd2: $t \in \mathbb{N}$
then
act1: $state := preempt$
act2: $s\_deliver\_time := t$
act3: $s\_deliver\_rate := 0$
end
Event resume_from_preempt ⟨ordinary⟩ $\widehat{=}$
extends resume_from_preempt
when
grd1: $state = preempt$
then
act1: $state := deliver$
act3: $s\_deliver\_rate := s\_r$
end
END

MACHINE Square_Dual_bolus2
REFINES Square1
SEES c_sd_bolus,c_normalbolus
VARIABLES

state
s_r
s_t
d_deliver_time
d_deliver_rate
d_t
sd_module
sd_flag

INVARIANTS

inv1: $d\_deliver\_time \in \mathbb{N}$
inv2: $d\_deliver\_rate \in \mathbb{N}$
inv3: $d\_t \in \mathbb{N}$
inv4: $sd\_module \in SDF$
inv5: $sd\_flag \in SDF$
inv6: $state = off \lor state = suspend \Rightarrow d\_deliver\_time = 0 \land d\_deliver\_rate = 0$
inv7: $state = deliver \Rightarrow d\_deliver\_time \geq 0 \land d\_deliver\_rate > 0$
inv8: $state = off \lor state = suspend \Rightarrow d\_t = 0$
inv9: $state = deliver \lor state = preempt \Rightarrow s\_r > 0$
inv10: $state = preempt \Rightarrow d\_deliver\_time \geq 0 \land d\_deliver\_rate = 0$
inv11: $sd\_module = d \land sd\_flag = s \land state = deliver \Rightarrow d\_deliver\_rate = s\_r$
inv12: $sd\_module = d \land sd\_flag = d \land state = deliver \Rightarrow d\_deliver\_rate = normal\_bolus\_rate$
inv13: $sd\_module = s \land state = deliver \Rightarrow d\_deliver\_rate = s\_r$
inv14: $state = off \lor state = suspend \Rightarrow sd\_flag = d$
Square1.inv1: $s\_r \in \mathbb{N}$
Square1.inv2: $s\_t \in \mathbb{N}$
Square1.inv3: $s\_deliver\_time \in \mathbb{N}$
Square1.inv4: $s\_deliver\_rate \in \mathbb{N}$
Square1.inv8: $state = off \lor state = suspend \Rightarrow s\_r = 0 \land s\_t = 0$
Square1.inv10: $state = deliver \lor state = preempt \Rightarrow s\_t > 0$
Square.inv1: $state \in SD$

EVENTS
Initialisation

begin

act1: $state := off$
act2: $s\_r := 0$
act3: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_module := s$
act10: $sd\_flag := d$

end

Event start ⟨ordinary⟩ $\widehat{=}$
refines start

any

t
r

where

grd1: $state = off$
grd2: $t \in \mathbb{N}_1$

grd3: $r \in \mathbb{N}_1$

then

act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act6: $sd\_module := s$
act7: $d\_deliver\_time := t$
act8: $d\_deliver\_rate := r$

end

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

refines start

any

t
r
td

where

grd1: $state = off$
grd2: $t \in \mathbb{N}_1$
grd3: $r \in \mathbb{N}_1$
grd4: $td \in \mathbb{N}_1$

then

act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act6: $d\_deliver\_time := t + td$
act7: $d\_deliver\_rate := normal\_bolus\_rate$
act8: $d\_t := td$
act9: $sd\_module := d$

end

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

when

grd2: $state = deliver$
grd3: $sd\_module = d$
grd4: $sd\_flag = d$

then

act2: $d\_deliver\_rate := s\_r$
act3: $sd\_flag := s$

end

Event finish ⟨ordinary⟩ $\widehat{=}$

refines finish

when

grd1: $state = deliver$
grd2: $sd\_module = d \Rightarrow sd\_flag = s$

then

act1: $state := off$
act4: $s\_r := 0$
act5: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_flag := d$

end

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

refines suspend

when

grd1: $state = deliver$

then

act1: $state := suspend$

act4: $s\_r := 0$
act5: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_flag := d$

end

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

when

grd1: $state = suspend$

then

act1: $state := off$

end

Event preempted ⟨ordinary⟩ $\widehat{=}$
refines preempted

any

t time left for square bolus

where

grd1: $state = deliver$
grd2: $t \in 0 \mathinner{..} d\_deliver\_time$

then

act1: $state := preempt$
act4: $d\_deliver\_time := t$
act5: $d\_deliver\_rate := 0$

end

Event resume_from_preempt ⟨ordinary⟩ $\widehat{=}$
refines resume_from_preempt

any

r

where

grd1: $state = preempt$
grd2: $sd\_module = s \Rightarrow r = s\_r$
grd3: $sd\_module = d \wedge sd\_flag = d \Rightarrow r = normal\_bolus\_rate$
grd4: $sd\_module = d \wedge sd\_flag = s \Rightarrow r = s\_r$

then

act1: $state := deliver$
act4: $d\_deliver\_rate := r$

end

END

MACHINE Square_Dual_bolus2_continuous
REFINES Square_Dual_bolus2
SEES c_sd_bolus,c_normalbolus
VARIABLES

state
s_r
s_t
d_deliver_time
d_deliver_rate
d_t
sd_module
sd_flag
sd_now
sd_new_now
sd_rate_c

INVARIANTS

inv5: $sd\_rate\_c \in \mathbb{N} \pfun \mathbb{N}$
inv1: $sd\_now \in dom(sd\_rate\_c)$
inv2: $sd\_new\_now \in \mathbb{N}$
inv4: $d\_deliver\_rate = sd\_rate\_c(sd\_now)$
inv6: $state \in \{deliver, preempt\} \land sd\_module = d \land sd\_flag = d \Rightarrow d\_deliver\_time > s\_t$

EVENTS

Initialisation ⟨extended⟩

begin

act1: $state := off$
act2: $s\_r := 0$
act3: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_module := s$
act10: $sd\_flag := d$
act11: $sd\_now := 0$
act12: $sd\_new\_now := 0$
act13: $sd\_rate\_c := \{0 \mapsto 0\}$

end

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

any

t
r
ctime

where

grd1: $state = off$
grd2: $t \in \mathbb{N}_1$
grd3: $r \in \mathbb{N}_1$
grd4: $ctime \geq sd\_now$

then

act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act6: $sd\_module := s$
act7: $d\_deliver\_time := t$
act8: $d\_deliver\_rate := r$
act9: $sd\_rate\_c := \lambda x \cdot x \in ctime \mathinner{..} ctime + t | r$
act10: $sd\_now := ctime$

act11: $sd\_new\_now := ctime + t$

end

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

extends start_dual

any

t

r

td

ctime

where

grd1: $state = off$

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

grd3: $r \in \mathbb{N}_1$

grd4: $td \in \mathbb{N}_1$

grd5: $ctime \geq sd\_now$

then

act1: $state := deliver$

act2: $s\_r := r$

act3: $s\_t := t$

act6: $d\_deliver\_time := t + td$

act7: $d\_deliver\_rate := normal\_bolus\_rate$

act8: $d\_t := td$

act9: $sd\_module := d$

act11: $sd\_rate\_c := \lambda x \cdot x \in ctime \mathinner{..} ctime + td | normal\_bolus\_rate$

act12: $sd\_now := ctime$

act10: $sd\_new\_now := ctime + td$

end

Event finish ⟨ordinary⟩ $\widehat{=}$

extends finish

any

ctime

where

grd1: $state = deliver$

grd2: $sd\_module = d \Rightarrow sd\_flag = s$

grd3: $ctime = sd\_new\_now$

then

act1: $state := off$

act4: $s\_r := 0$

act5: $s\_t := 0$

act6: $d\_deliver\_time := 0$

act7: $d\_deliver\_rate := 0$

act8: $d\_t := 0$

act9: $sd\_flag := d$

act10: $sd\_rate\_c := \lambda x \cdot x \geq ctime | 0$

act11: $sd\_now := ctime$

end

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

extends suspend

any

ctime

where

grd1: $state = deliver$

grd2: $ctime \in sd\_now \mathinner{..} sd\_new\_now$

then

act1: $state := suspend$

act4: $s\_r := 0$

act5: $s\_t := 0$

act6: $d\_deliver\_time := 0$

act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_flag := d$
act10: $sd\_now := ctime$
act11: $sd\_rate\_c := \lambda x \cdot x \geq ctime | 0$

end

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

any
ctime

where
grd1: $state = suspend$
grd2: $ctime > sd\_now$

then
act1: $state := off$
act2: $sd\_now := ctime$
act3: $sd\_rate\_c := \lambda x \cdot x \geq ctime | 0$

end

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

any
t time left for square bolus
ctime

where
grd1: $state = deliver$
grd2: $t \in 0 \,..\, d\_deliver\_time$
grd3: $ctime \in sd\_now \,..\, sd\_new\_now$
grd4: $sd\_module = d \wedge sd\_flag = d \Rightarrow t \in s\_t + 1 \,..\, d\_deliver\_time$
grd5: $sd\_module = d \wedge sd\_flag = s \Rightarrow t \in 0 \,..\, s\_t$

then
act1: $state := preempt$
act4: $d\_deliver\_time := t$
act5: $d\_deliver\_rate := 0$
act6: $sd\_now := ctime$
act7: $sd\_rate\_c := \lambda x \cdot x \geq ctime | 0$

end

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

any
r
ctime
t2

where
grd1: $state = preempt$
grd2: $sd\_module = s \Rightarrow r = s\_r$
grd3: $sd\_module = d \wedge sd\_flag = d \Rightarrow r = normal\_bolus\_rate$
grd4: $sd\_module = d \wedge sd\_flag = s \Rightarrow r = s\_r$
grd9: $ctime \in \mathbb{N}$
grd5: $ctime > sd\_now$
grd10: $t2 \in \mathbb{N}$
grd6: $sd\_module = s \Rightarrow t2 = ctime + d\_deliver\_time$
grd7: $sd\_module = d \wedge sd\_flag = d \Rightarrow t2 = ctime + d\_deliver\_time - s\_t$
grd8: $sd\_module = d \wedge sd\_flag = s \Rightarrow t2 = ctime + d\_deliver\_time$

then
act1: $state := deliver$
act4: $d\_deliver\_rate := r$
act5: $sd\_now := ctime$
act6: $sd\_rate\_c := \lambda x \cdot x \in ctime \,..\, t2 | r$

act7: $sd\_new\_now := t2$

end

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

extends update_to_dual

any

ctime

where

grd2: $state = deliver$

grd3: $sd\_module = d$

grd4: $sd\_flag = d$

grd5: $ctime = sd\_new\_now$

then

act2: $d\_deliver\_rate := s\_r$

act3: $sd\_flag := s$

act4: $sd\_now := ctime$

act5: $sd\_new\_now := ctime + s\_t$

act6: $sd\_rate\_c := \lambda x \cdot x \in ctime \mathinner{..} ctime + s\_t | s\_r$

end

END

MACHINE Square_Dual_bolus2_continuous_2
REFINES Square_Dual_bolus2_continuous
SEES c_sd_bolus,c_normalbolus
VARIABLES

state
s_r
s_t
d_deliver_time
d_deliver_rate
d_t
sd_module
sd_flag
sd_now
sd_new_now
sd_rate_c
time
t_sd

INVARIANTS

inv1: $time \in \mathbb{N}$
inv3: $t\_sd \in \mathbb{N}$
inv4: $state = preempt \Rightarrow t\_sd \leq d\_t + s\_t$
inv5: $state = deliver \Rightarrow t\_sd \leq time + d\_t + s\_t$
inv6: $state = deliver \Rightarrow t\_sd \geq time$
inv8: $state = deliver \Rightarrow t\_sd - time \leq d\_deliver\_time$
inv7: $state \in \{deliver\} \land sd\_module = d \land sd\_flag = s \Rightarrow sd\_new\_now = t\_sd$
inv11: $state \in \{deliver\} \land sd\_module = s \Rightarrow sd\_new\_now = t\_sd$
inv2: ⟨theorem⟩ $state = deliver \Rightarrow t\_sd - time \in 0 \mathinner{..} d\_deliver\_time$
inv9: $state \in \{deliver\} \land sd\_module = d \land sd\_flag = d \Rightarrow sd\_new\_now + s\_t = t\_sd$
inv10: $state = preempt \Rightarrow d\_deliver\_time = t\_sd$
inv12: $state = deliver \land sd\_module = d \land sd\_flag = d \Rightarrow time \leq sd\_new\_now$

EVENTS

Initialisation ⟨extended⟩

begin

act1: $state := off$
act2: $s\_r := 0$
act3: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_module := s$
act10: $sd\_flag := d$
act11: $sd\_now := 0$
act12: $sd\_new\_now := 0$
act13: $sd\_rate\_c := \{0 \mapsto 0\}$
act14: $time := 0$
act15: $t\_sd := 0$

end

Event start ⟨ordinary⟩ $\widehat{=}$
refines start

any

t
r

where

grd1: $state = off$

grd2: $t \in \mathbb{N}_1$
grd3: $r \in \mathbb{N}_1$
grd4: $time \geq sd\_now$

with

ctime: $ctime = time$

then

act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act6: $sd\_module := s$
act7: $d\_deliver\_time := t$
act8: $d\_deliver\_rate := r$
act9: $sd\_rate\_c := \lambda x \cdot x \in time \mathinner{..} time + t | r$
act10: $sd\_now := time$
act11: $sd\_new\_now := time + t$
act12: $t\_sd := time + t$

end

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

refines start_dual

any

t
r
td

where

grd1: $state = off$
grd2: $t \in \mathbb{N}_1$
grd3: $r \in \mathbb{N}_1$
grd4: $td \in \mathbb{N}_1$
grd5: $time \geq sd\_now$

with

ctime: $ctime = time$

then

act1: $state := deliver$
act2: $s\_r := r$
act3: $s\_t := t$
act6: $d\_deliver\_time := t + td$
act7: $d\_deliver\_rate := normal\_bolus\_rate$
act8: $d\_t := td$
act9: $sd\_module := d$
act11: $sd\_rate\_c := \lambda x \cdot x \in time \mathinner{..} time + td | normal\_bolus\_rate$
act12: $sd\_now := time$
act10: $sd\_new\_now := time + td$
act13: $t\_sd := time + t + td$

end

Event finish ⟨ordinary⟩ $\widehat{=}$

refines finish

when

grd1: $state = deliver$
grd2: $sd\_module = d \Rightarrow sd\_flag = s$
grd3: $time = sd\_new\_now$
grd4: ⟨theorem⟩ $time = t\_sd$

with

ctime: $ctime = time$

then

act1: $state := off$
act4: $s\_r := 0$
act5: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$

act8: $d\_t := 0$
act9: $sd\_flag := d$
act10: $sd\_rate\_c := \lambda x \cdot x \geq time | 0$
act11: $sd\_now := time$

end

Event suspend ⟨ordinary⟩ $\widehat{=}$
refines suspend

when

grd1: $state = deliver$
grd2: $time \in sd\_now \mathinner{..} sd\_new\_now$

with

ctime: $ctime = time$

then

act1: $state := suspend$
act4: $s\_r := 0$
act5: $s\_t := 0$
act6: $d\_deliver\_time := 0$
act7: $d\_deliver\_rate := 0$
act8: $d\_t := 0$
act9: $sd\_flag := d$
act10: $sd\_now := time$
act11: $sd\_rate\_c := \lambda x \cdot x \geq time | 0$

end

Event resume ⟨ordinary⟩ $\widehat{=}$
refines resume

when

grd1: $state = suspend$
grd2: $time > sd\_now$

with

ctime: $ctime = time$

then

act1: $state := off$
act2: $sd\_now := time$
act3: $sd\_rate\_c := \lambda x \cdot x \geq time | 0$

end

Event preempted ⟨ordinary⟩ $\widehat{=}$
refines preempted

when

grd1: $state = deliver$
grd3: $time \in sd\_now \mathinner{..} sd\_new\_now$
grd4: $sd\_module = d \wedge sd\_flag = d \Rightarrow t\_sd - time \in s\_t + 1 \mathinner{..} d\_deliver\_time$
grd5: $sd\_module = d \wedge sd\_flag = s \Rightarrow t\_sd - time \in 0 \mathinner{..} s\_t$

with

ctime: $ctime = time$
t: $t = t\_sd - time$

then

act1: $state := preempt$
act4: $d\_deliver\_time := t\_sd - time$
act5: $d\_deliver\_rate := 0$
act6: $sd\_now := time$
act7: $sd\_rate\_c := \lambda x \cdot x \geq time | 0$
act8: $t\_sd := t\_sd - time$

end

Event resume_from_preempt ⟨ordinary⟩ $\widehat{=}$
refines resume_from_preempt

any

r
t2

where

grd1: $state = preempt$
grd2: $sd\_module = s \Rightarrow r = s\_r$
grd3: $sd\_module = d \land sd\_flag = d \Rightarrow r = normal\_bolus\_rate$
grd4: $sd\_module = d \land sd\_flag = s \Rightarrow r = s\_r$
grd5: $time > sd\_now$
grd6: $sd\_module = s \Rightarrow t2 = time + d\_deliver\_time$
grd7: $sd\_module = d \land sd\_flag = d \Rightarrow t2 = time + d\_deliver\_time - s\_t$
grd8: $sd\_module = d \land sd\_flag = s \Rightarrow t2 = time + d\_deliver\_time$

with

ctime: $ctime = time$

then

act1: $state := deliver$
act4: $d\_deliver\_rate := r$
act5: $sd\_now := time$
act6: $sd\_rate\_c := \lambda x \cdot x \in time \mathinner{..} t2 | r$
act7: $sd\_new\_now := t2$
act8: $t\_sd := time + t\_sd$

end

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

refines update_to_dual

when

grd2: $state = deliver$
grd3: $sd\_module = d$
grd4: $sd\_flag = d$
grd5: $time = sd\_new\_now$

with

ctime: $ctime = time$

then

act2: $d\_deliver\_rate := s\_r$
act3: $sd\_flag := s$
act4: $sd\_now := time$
act5: $sd\_new\_now := time + s\_t$
act6: $sd\_rate\_c := \lambda x \cdot x \in time \mathinner{..} time + s\_t | s\_r$

end

Event timer ⟨ordinary⟩ $\widehat{=}$

when

grd1:
$\neg((state = deliver \land (sd\_module = d \Rightarrow sd\_flag = s) \land time = sd\_new\_now \land time = t\_sd) \lor$
$(state = deliver \land sd\_module = d \land sd\_flag = d \land time = sd\_new\_now))$

then

act1: $time := time + 1$

end

END

MACHINE Basal
SEES c_basal
VARIABLES
rate_setting
basal_rate_in
basal_mode
INVARIANTS
inv1: $rate\_setting \in 0 \mathinner{..} c-1 \pfun 0 \mathinner{..} basal\_max \land 0 \in dom(rate\_setting)$
inv2: $basal\_rate\_in \in 0 \mathinner{..} basal\_max$
inv3: $basal\_mode \in BASALMODE$
EVENTS
Initialisation
begin
act1: $rate\_setting := \{0 \mapsto 0\}$
act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
end
Event basal_suspend ⟨ordinary⟩ $\widehat{=}$
when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
end
Event basal_resume ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd1: $basal\_rate\_in = 0$
grd2: $t \in 0 \mathinner{..} c-1$
grd3: $basal\_mode = suspended$
then
act1: $basal\_rate\_in := rate\_setting(max(\{i \mid i \in dom(rate\_setting) \land i \leq t\}))$
act2: $basal\_mode := delivering$
end
Event change_setting ⟨ordinary⟩ $\widehat{=}$
any
t
r
where
grd1: $t \in dom(rate\_setting)$
grd2: $r \in 0 \mathinner{..} basal\_max$
then
act1: $rate\_setting := rate\_setting \rhd\!\!+ \{t \mapsto r\}$
end
Event delete_setting ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd1: $t \in dom(rate\_setting) \setminus \{0\}$
grd2: $basal\_mode \neq suspended$
then
act1: $rate\_setting := \{t\} \rhd\!\!- rate\_setting$
end
Event add_setting ⟨ordinary⟩ $\widehat{=}$

any
t
r
where
grd1: $t \notin dom(rate\_setting)$
grd5: $t \in 0 \mathinner{..} c - 1$
grd3: $r \in 0 \mathinner{..} basal\_max$
grd4: $basal\_mode \neq suspended$
then
act1: $rate\_setting := rate\_setting \cup \{t \mapsto r\}$
end

Event rate_update ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd1: $basal\_mode = delivering$
grd2: $t \in dom(rate\_setting)$
then
act1: $basal\_rate\_in := rate\_setting(t)$
end

Event start ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd1: $basal\_mode = stop$
grd2: $t \in 0 \mathinner{..} c - 1$
then
act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := rate\_setting(max(\{i | i \in dom(rate\_setting) \wedge i \leq t\}))$
end

Event stop ⟨ordinary⟩ $\widehat{=}$
when
grd1: $basal\_mode = delivering$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
end

END

MACHINE Basal1
REFINES Basal
SEES c_basal
VARIABLES
rate_setting
basal_rate_in
basal_mode
btime
INVARIANTS
inv1: $btime \in 1 \mathinner{..} c$
EVENTS
Initialisation ⟨extended⟩
begin
act1: $rate\_setting := \{0 \mapsto 0\}$
act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
end
Event basal_suspend ⟨ordinary⟩ $\widehat{=}$
extends basal_suspend
when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
end
Event basal_resume ⟨ordinary⟩ $\widehat{=}$
extends basal_resume
any
t
t2
where
grd1: $basal\_rate\_in = 0$
grd2: $t \in 0 \mathinner{..} c - 1$
grd3: $basal\_mode = suspended$
grd4: $\{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing \Rightarrow t2 = c$
grd5: $\{i | i \in dom(rate\_setting) \wedge i > t\} \neq \varnothing \Rightarrow t2 = min(\{i | i \in dom(rate\_setting) \wedge i > t\})$
then
act1: $basal\_rate\_in := rate\_setting(max(\{i | i \in dom(rate\_setting) \wedge i \leq t\}))$
act2: $basal\_mode := delivering$
act3: $btime := t2 - t$
end
Event change_setting ⟨ordinary⟩ $\widehat{=}$
extends change_setting
any
t
r
where
grd1: $t \in dom(rate\_setting)$
grd2: $r \in 0 \mathinner{..} basal\_max$
then
act1: $rate\_setting := rate\_setting \lhd\!\!\!\!- \{t \mapsto r\}$
end
Event delete_setting ⟨ordinary⟩ $\widehat{=}$
extends delete_setting

any
t
where
grd1: $t \in dom(rate\_setting) \setminus \{0\}$
grd2: $basal\_mode \neq suspended$
then
act1: $rate\_setting := \{t\} \lhd rate\_setting$
end
Event add_setting ⟨ordinary⟩ $\widehat{=}$
extends add_setting
any
t
r
where
grd1: $t \notin dom(rate\_setting)$
grd5: $t \in 0 \mathinner{..} c-1$
grd3: $r \in 0 \mathinner{..} basal\_max$
grd4: $basal\_mode \neq suspended$
then
act1: $rate\_setting := rate\_setting \cup \{t \mapsto r\}$
end
Event rate_update ⟨ordinary⟩ $\widehat{=}$
extends rate_update
any
t
t2
where
grd1: $basal\_mode = delivering$
grd2: $t \in dom(rate\_setting)$
grd3: $\{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing \Rightarrow t2 = c$
grd4: $\{i | i \in dom(rate\_setting) \wedge i > t\} \neq \varnothing \Rightarrow t2 = min(\{i | i \in dom(rate\_setting) \wedge i > t\})$
then
act1: $basal\_rate\_in := rate\_setting(t)$
act2: $btime := t2 - t$
end
Event start ⟨ordinary⟩ $\widehat{=}$
extends start
any
t
t2
where
grd1: $basal\_mode = stop$
grd2: $t \in 0 \mathinner{..} c-1$
grd3: $\{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing \Rightarrow t2 = c$
grd4: $\{i | i \in dom(rate\_setting) \wedge i > t\} \neq \varnothing \Rightarrow t2 = min(\{i | i \in dom(rate\_setting) \wedge i > t\})$
then
act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := rate\_setting(max(\{i | i \in dom(rate\_setting) \wedge i \leq t\}))$
act3: $btime := t2 - t$
end
Event stop ⟨ordinary⟩ $\widehat{=}$
extends stop
when
grd1: $basal\_mode = delivering$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
end
END

MACHINE Basal2
REFINES Basal1
SEES c_basal
VARIABLES

rate_setting
basal_rate_in
basal_mode
btime
rate_setting2

INVARIANTS

inv1: $rate\_setting2 \in 0 \mathinner{..} c-1 \rightarrow 0 \mathinner{..} basal\_max \cup \{-1\}$
inv2: $rate\_setting \subseteq rate\_setting2$
inv3: $rate\_setting2 \rhd \{-1\} = rate\_setting$
inv4: ⟨theorem⟩ $\forall t \cdot t \in 0 \mathinner{..} c-1 \Rightarrow ((\forall j \cdot j \in dom(rate\_setting2) \wedge j > t \Rightarrow rate\_setting2(j) = -1) \Leftrightarrow \{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing)$

EVENTS
Initialisation ⟨extended⟩

begin
act1: $rate\_setting := \{0 \mapsto 0\}$
act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \mathinner{..} c-1 \times \{-1\}) \cup \{0 \mapsto 0\}$
end

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

when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
end

Event basal_resume ⟨ordinary⟩ $\widehat{=}$
refines basal_resume

any
t
t2
where
grd1: $basal\_rate\_in = 0$
grd2: $t \in 0 \mathinner{..} c-1$
grd3: $basal\_mode = suspended$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \wedge j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \wedge j > t \wedge rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \wedge i > t\})$
grd8: ⟨theorem⟩ $(\forall j \cdot j \in dom(rate\_setting2) \wedge j > t \Rightarrow rate\_setting2(j) = -1) \Leftrightarrow \{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing$
grd9: ⟨theorem⟩ $(\exists j \cdot j \in dom(rate\_setting2) \wedge j > t \wedge rate\_setting2(j) \neq -1) \Leftrightarrow \{i | i \in dom(rate\_setting) \wedge i > t\} \neq \varnothing$
grd4: ⟨theorem⟩ $\{i | i \in dom(rate\_setting) \wedge i > t\} = \varnothing \Rightarrow t2 = c$
grd5: ⟨theorem⟩ $\{i | i \in dom(rate\_setting) \wedge i > t\} \neq \varnothing \Rightarrow t2 = min(\{i | i \in dom(rate\_setting) \wedge i > t\})$
then
act1: $basal\_rate\_in := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \wedge i \leq t\}))$
act2: $basal\_mode := delivering$
act3: $btime := t2 - t$
end

Event change_setting ⟨ordinary⟩ $\widehat{=}$
refines change_setting
any
t
r
where
grd1: $t \in dom(rate\_setting2 \rhd \{-1\})$
grd2: $r \in 0 \mathinner{..} basal\_max$
then
act1: $rate\_setting := rate\_setting \lhd\!\!\!- \{t \mapsto r\}$
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto r\}$
end
Event delete_setting ⟨ordinary⟩ $\widehat{=}$
refines delete_setting
any
t
where
grd1: $t \in dom(rate\_setting2 \rhd \{-1\}) \setminus \{0\}$
grd2: $basal\_mode \neq suspended$
then
act1: $rate\_setting := \{t\} \lhd rate\_setting$
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto -1\}$
end
Event add_setting ⟨ordinary⟩ $\widehat{=}$
refines add_setting
any
t
r
where
grd3: $r \in 0 \mathinner{..} basal\_max$
grd4: $basal\_mode \neq suspended$
grd5: $t \in 0 \mathinner{..} c - 1$
grd6: $rate\_setting2(t) = -1$
grd1: ⟨theorem⟩ $t \notin dom(rate\_setting)$
then
act1: $rate\_setting := rate\_setting \cup \{t \mapsto r\}$
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto r\}$
end
Event rate_update ⟨ordinary⟩ $\widehat{=}$
refines rate_update
any
t
t2
where
grd1: $basal\_mode = delivering$
grd11: $t \in dom(rate\_setting2 \rhd \{-1\})$
grd2: ⟨theorem⟩ $t \in dom(rate\_setting)$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
grd9: ⟨theorem⟩ $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Leftrightarrow \{i \mid i \in dom(rate\_setting) \land i > t\} \neq \varnothing$
grd10: ⟨theorem⟩ $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Leftrightarrow \{i \mid i \in dom(rate\_setting) \land i > t\} = \varnothing$
grd3: ⟨theorem⟩ $\{i \mid i \in dom(rate\_setting) \land i > t\} = \varnothing \Rightarrow t2 = c$
grd4: ⟨theorem⟩ $\{i \mid i \in dom(rate\_setting) \land i > t\} \neq \varnothing \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting) \land i > t\})$
then
act1: $basal\_rate\_in := rate\_setting2(t)$

act2: $btime := t2 - t$

end

Event start ⟨ordinary⟩ $\widehat{=}$

refines start

any

t

t2

where

grd1: $basal\_mode = stop$

grd2: $t \in 0 \mathinner{..} c - 1$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$

grd10: ⟨theorem⟩ $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Leftrightarrow \{i | i \in dom(rate\_setting) \land i > t\} = \varnothing$

grd9: ⟨theorem⟩ $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Leftrightarrow \{i | i \in dom(rate\_setting) \land i > t\} \neq \varnothing$

grd3: ⟨theorem⟩ $\{i | i \in dom(rate\_setting) \land i > t\} = \varnothing \Rightarrow t2 = c$

grd4: ⟨theorem⟩ $\{i | i \in dom(rate\_setting) \land i > t\} \neq \varnothing \Rightarrow t2 = min(\{i | i \in dom(rate\_setting) \land i > t\})$

then

act1: $basal\_mode := delivering$

act2: $basal\_rate\_in := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$

act3: $btime := t2 - t$

end

Event stop ⟨ordinary⟩ $\widehat{=}$

extends stop

when

grd1: $basal\_mode = delivering$

then

act1: $basal\_mode := stop$

act2: $basal\_rate\_in := 0$

end

END

MACHINE Basal3
REFINES Basal2
SEES c_basal
VARIABLES
basal_rate_in
basal_mode
btime
rate_setting2
EVENTS
Initialisation
begin
act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \mathinner{..} c-1 \times \{-1\}) \cup \{0 \mapsto 0\}$
end
Event basal_suspend ⟨ordinary⟩ $\widehat{=}$
extends basal_suspend
when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
end
Event basal_resume ⟨ordinary⟩ $\widehat{=}$
refines basal_resume
any
t
t2
where
grd1: $basal\_rate\_in = 0$
grd2: $t \in 0 \mathinner{..} c-1$
grd3: $basal\_mode = suspended$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
then
act1: $basal\_rate\_in := rate\_setting2(max(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$
act2: $basal\_mode := delivering$
act3: $btime := t2 - t$
end
Event change_setting ⟨ordinary⟩ $\widehat{=}$
refines change_setting
any
t
r
where
grd1: $t \in dom(rate\_setting2 \rhd \{-1\})$
grd2: $r \in 0 \mathinner{..} basal\_max$
then
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!\!- \{t \mapsto r\}$
end
Event delete_setting ⟨ordinary⟩ $\widehat{=}$
refines delete_setting
any

t

where

grd1: $t \in dom(rate\_setting2 \rhd \{-1\}) \setminus \{0\}$

grd2: $basal\_mode \neq suspended$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto -1\}$

end

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

refines add_setting

any

t

r

where

grd3: $r \in 0 \mathinner{..} basal\_max$

grd4: $basal\_mode \neq suspended$

grd5: $t \in 0 \mathinner{..} c - 1$

grd6: $rate\_setting2(t) = -1$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto r\}$

end

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

refines rate_update

any

t

t2

where

grd1: $basal\_mode = delivering$

grd11: $t \in dom(rate\_setting2 \rhd \{-1\})$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \wedge j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \wedge j > t \wedge rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \wedge i > t\})$

then

act1: $basal\_rate\_in := rate\_setting2(t)$

act2: $btime := t2 - t$

end

Event start ⟨ordinary⟩ $\widehat{=}$

refines start

any

t

t2

where

grd1: $basal\_mode = stop$

grd2: $t \in 0 \mathinner{..} c - 1$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \wedge j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \wedge j > t \wedge rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \wedge i > t\})$

then

act1: $basal\_mode := delivering$

act2: $basal\_rate\_in := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \wedge i \leq t\}))$

act3: $btime := t2 - t$

end

Event stop ⟨ordinary⟩ $\widehat{=}$

extends stop

when

grd1: $basal\_mode = delivering$

then

act1: $basal\_mode := stop$

act2: $basal\_rate\_in := 0$

end

END

MACHINE Basal4
REFINES Basal3
SEES c_basal
VARIABLES
basal_rate_in
basal_mode
btime
rate_setting2
min_value
max_value
INVARIANTS
inv1: $min\_value \in 0 \mathinner{..} c$
inv2: $max\_value \in 0 \mathinner{..} basal\_max$
EVENTS
Initialisation ⟨extended⟩
begin
act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \mathinner{..} c - 1 \times \{-1\}) \cup \{0 \mapsto 0\}$
act6: $min\_value := 0$
act7: $max\_value := 0$
end
Event basal_suspend ⟨ordinary⟩ $\widehat{=}$
extends basal_suspend
when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
end
Event basal_resume ⟨ordinary⟩ $\widehat{=}$
extends basal_resume
any
t
$t2$
where
grd1: $basal\_rate\_in = 0$
grd2: $t \in 0 \mathinner{..} c - 1$
grd3: $basal\_mode = suspended$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
then
act1: $basal\_rate\_in := rate\_setting2(max(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$
act2: $basal\_mode := delivering$
act3: $btime := t2 - t$
end
Event change_setting ⟨ordinary⟩ $\widehat{=}$
extends change_setting
any
t
r
where
grd1: $t \in dom(rate\_setting2 \rhd \{-1\})$

grd2: $r \in 0 \mathinner{..} basal\_max$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto r\}$

end

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

extends delete_setting

any

t

where

grd1: $t \in dom(rate\_setting2 \rhd \{-1\}) \setminus \{0\}$

grd2: $basal\_mode \neq suspended$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto -1\}$

end

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

extends add_setting

any

t

r

where

grd3: $r \in 0 \mathinner{..} basal\_max$

grd4: $basal\_mode \neq suspended$

grd5: $t \in 0 \mathinner{..} c - 1$

grd6: $rate\_setting2(t) = -1$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!- \{t \mapsto r\}$

end

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

extends rate_update

any

t

$t2$

where

grd1: $basal\_mode = delivering$

grd11: $t \in dom(rate\_setting2 \rhd \{-1\})$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$

then

act1: $basal\_rate\_in := rate\_setting2(t)$

act2: $btime := t2 - t$

end

Event start ⟨ordinary⟩ $\widehat{=}$

extends start

any

t

$t2$

where

grd1: $basal\_mode = stop$

grd2: $t \in 0 \mathinner{..} c - 1$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$

then

act1: $basal\_mode := delivering$

act2: $basal\_rate\_in := rate\_setting2(max(\{i \mid i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$

act3: $btime := t2 - t$

end

Event stop ⟨ordinary⟩ $\widehat{=}$
extends stop
when
grd1: $basal\_mode = delivering$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
end
Event get_min_value_1 ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd5: $t \in 0 \mathinner{..} c - 1$
grd3: $\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1$
then
act1: $min\_value := c$
end
Event get_min_value_2 ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd5: $t \in 0 \mathinner{..} c - 1$
grd4: $\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1$
then
act1: $min\_value := min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
end
Event get_max_value ⟨ordinary⟩ $\widehat{=}$
any
t
where
grd1: $t \in 0 \mathinner{..} c - 1$
then
act1: $max\_value := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$
end
END

MACHINE Basal5
REFINES Basal4
SEES c_basal
VARIABLES

basal_rate_in
basal_mode
btime
rate_setting2
min_value
get_min_value_add
par_t
temp_min
get_min_start_t
max_value
get_max_start_t
get_max_value_add
par_t_max

INVARIANTS

inv1: $get\_min\_value\_add \in 0 \mathinner{..} 3$
inv2: $par\_t \in \mathbb{N}$
inv3: $temp\_min \in 0 \mathinner{..} c$
inv4: $get\_min\_start\_t \in 0 \mathinner{..} c - 1$
inv7: $get\_min\_value\_add = 3 \Rightarrow \{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > get\_min\_start\_t\} \neq \varnothing$
inv5: $get\_min\_value\_add = 3 \Rightarrow temp\_min = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > get\_min\_start\_t\})$
inv8: $get\_min\_value\_add = 1 \Rightarrow par\_t = get\_min\_start\_t + 1$
inv10: $get\_min\_value\_add = 2 \Rightarrow par\_t > get\_min\_start\_t$
inv9: $get\_min\_value\_add = 2 \Rightarrow \{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > get\_min\_start\_t \land i \leq par\_t - 1\} = \varnothing$
inv11: $get\_max\_start\_t \in 0 \mathinner{..} c - 1$
inv12: $get\_max\_value\_add \in 0 \mathinner{..} 2$
inv13: $par\_t\_max \in 0 \mathinner{..} c - 1$
inv14: $get\_max\_value\_add \in \{1, 2\} \Rightarrow get\_max\_start\_t \in 0 \mathinner{..} c - 1$
inv15: $get\_max\_value\_add = 2 \Rightarrow par\_t\_max = max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq get\_max\_start\_t\})$
inv16: $get\_max\_value\_add = 1 \Rightarrow \{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq get\_max\_start\_t \land i \geq par\_t\_max + 1\} = \varnothing$
inv17: $get\_max\_value\_add \in \{1, 2\} \Rightarrow par\_t\_max \leq get\_max\_start\_t$

EVENTS

Initialisation ⟨extended⟩

begin

act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \mathinner{..} c - 1 \times \{-1\}) \cup \{0 \mapsto 0\}$
act6: $min\_value := 0$
act7: $max\_value := 0$
act11: $get\_min\_value\_add := 0$
act8: $par\_t := 0$
act9: $temp\_min := 0$
act10: $get\_min\_start\_t := 0$
act12: $get\_max\_start\_t := 0$
act13: $get\_max\_value\_add := 0$
act14: $par\_t\_max := 0$

end

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

extends basal_suspend

when

grd1: $basal\_rate\_in \neq 0$

grd2: $basal\_mode = delivering$

then

act1: $basal\_rate\_in := 0$

act2: $basal\_mode := suspended$

end

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

extends basal_resume

any

t

$t2$

where

grd1: $basal\_rate\_in = 0$

grd2: $t \in 0 \mathinner{..} c - 1$

grd3: $basal\_mode = suspended$

grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$

grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$

then

act1: $basal\_rate\_in := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$

act2: $basal\_mode := delivering$

act3: $btime := t2 - t$

end

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

extends change_setting

any

t

r

where

grd1: $t \in dom(rate\_setting2 \rhd \{-1\})$

grd2: $r \in 0 \mathinner{..} basal\_max$

grd3: $get\_min\_value\_add = 0$

grd4: $get\_max\_value\_add = 0$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!\!- \{t \mapsto r\}$

end

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

extends delete_setting

any

t

where

grd1: $t \in dom(rate\_setting2 \rhd \{-1\}) \setminus \{0\}$

grd2: $basal\_mode \neq suspended$

grd3: $get\_min\_value\_add = 0$

grd4: $get\_max\_value\_add = 0$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!\!- \{t \mapsto -1\}$

end

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

extends add_setting

any

t

r

where
grd3: $r \in 0 \mathinner{..} basal\_max$
grd4: $basal\_mode \neq suspended$
grd5: $t \in 0 \mathinner{..} c - 1$
grd6: $rate\_setting2(t) = -1$
grd7: $get\_min\_value\_add = 0$
grd8: $get\_max\_value\_add = 0$
then
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!\!- \{t \mapsto r\}$
end

Event rate_update ⟨ordinary⟩ $\widehat{=}$
extends rate_update
any
t
$t2$
where
grd1: $basal\_mode = delivering$
grd11: $t \in dom(rate\_setting2 \rhd \{-1\})$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
then
act1: $basal\_rate\_in := rate\_setting2(t)$
act2: $btime := t2 - t$
end

Event start ⟨ordinary⟩ $\widehat{=}$
extends start
any
t
$t2$
where
grd1: $basal\_mode = stop$
grd2: $t \in 0 \mathinner{..} c - 1$
grd6: $(\forall j \cdot j \in dom(rate\_setting2) \land j > t \Rightarrow rate\_setting2(j) = -1) \Rightarrow t2 = c$
grd7: $(\exists j \cdot j \in dom(rate\_setting2) \land j > t \land rate\_setting2(j) \neq -1) \Rightarrow t2 = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > t\})$
then
act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq t\}))$
act3: $btime := t2 - t$
end

Event stop ⟨ordinary⟩ $\widehat{=}$
extends stop
when
grd1: $basal\_mode = delivering$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
end

Event get_min_value_1 ⟨ordinary⟩ $\widehat{=}$
refines get_min_value_1
when
grd4: $get\_min\_value\_add = 2$
grd5: $par\_t = c$
grd3: ⟨theorem⟩ $\forall j \cdot j \in dom(rate\_setting2) \land j > get\_min\_start\_t \Rightarrow rate\_setting2(j) = -1$
with
t: $t = get\_min\_start\_t$
then

act1: $min\_value := c$
act2: $get\_min\_value\_add := 0$

end

Event get_min_value_2 ⟨ordinary⟩ $\widehat{=}$

refines get_min_value_2

when

grd5: $get\_min\_value\_add = 3$
grd4: ⟨theorem⟩ $\exists j \cdot j \in dom(rate\_setting2) \land j > get\_min\_start\_t \land rate\_setting2(j) \neq -1$

with

t: $t = get\_min\_start\_t$

then

act1: $min\_value := temp\_min$
act2: $get\_min\_value\_add := 0$

end

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

any

t

where

grd1: $t \in 0 \,..\, c - 1$
grd2: $get\_min\_value\_add = 0$

then

act1: $par\_t := t + 1$
act2: $get\_min\_value\_add := 1$
act3: $get\_min\_start\_t := t$

end

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

when

grd1: $par\_t < c$
grd2: $get\_min\_value\_add \in \{1, 2\}$
grd3: $rate\_setting2(par\_t) = -1$

then

act1: $par\_t := par\_t + 1$
act2: $get\_min\_value\_add := 2$

end

Event find_min_value_2 ⟨ordinary⟩ $\widehat{=}$

when

grd1: $par\_t < c$
grd2: $get\_min\_value\_add \in \{1, 2\}$
grd3: $rate\_setting2(par\_t) \neq -1$

then

act1: $temp\_min := par\_t$
act2: $get\_min\_value\_add := 3$

end

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

refines get_max_value

when

grd2: $get\_max\_value\_add = 2$

with

t: $t = get\_max\_start\_t$

then

act1: $max\_value := rate\_setting2(par\_t\_max)$
act2: $get\_max\_value\_add := 0$

end

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

any

t

where

grd1: $t \in 0 \,..\, c - 1$

grd2: $get\_max\_value\_add = 0$

then

act1: $get\_max\_start\_t := t$

act2: $get\_max\_value\_add := 1$

act3: $par\_t\_max := t$

end

Event get_max_value_1 ⟨ordinary⟩ $\widehat{=}$

when

grd1: $get\_max\_value\_add = 1$

grd3: $par\_t\_max \geq 0$

grd2: $rate\_setting2(par\_t\_max) = -1$

then

act1: $par\_t\_max := par\_t\_max - 1$

end

Event get_max_value_2 ⟨ordinary⟩ $\widehat{=}$

when

grd1: $get\_max\_value\_add = 1$

grd2: ⟨theorem⟩ $par\_t\_max \geq 0$

grd3: $rate\_setting2(par\_t\_max) \neq -1$

then

act1: $get\_max\_value\_add := 2$

end

END

MACHINE Basal6
REFINES Basal5
SEES c_basal2
VARIABLES

basal_rate_in
basal_mode
btime
rate_setting2
min_value
get_min_value_add
par_t
temp_min
get_min_start_t
max_value
get_max_start_t
get_max_value_add
par_t_max
prog_basal
par_get_t
add_resume
add_update
add_start

INVARIANTS

inv1: $prog\_basal \in PROC\_BASAL$
inv2: $prog\_basal = null \Rightarrow get\_max\_value\_add = 0 \land get\_min\_value\_add = 0$
inv3: $par\_get\_t \in 0 \,..\, c-1$
inv13: $add\_resume \in 0 \,..\, 3$
inv14: $add\_update \in 0 \,..\, 3$
inv15: $add\_start \in 0 \,..\, 3$
inv17: $prog\_basal = null \Rightarrow add\_resume = 0 \land add\_update = 0 \land add\_start = 0$
inv18: $add\_resume \neq 0 \Rightarrow add\_update = 0 \land add\_start = 0$
inv26: $add\_start \neq 0 \Rightarrow add\_update = 0 \land add\_resume = 0$
inv27: $add\_update \neq 0 \Rightarrow add\_resume = 0 \land add\_start = 0$
inv11: $get\_max\_value\_add \neq 0 \Rightarrow prog\_basal = call\_get\_max$
inv12: $get\_min\_value\_add \neq 0 \Rightarrow prog\_basal = call\_get\_min$
inv7: $prog\_basal \in \{call\_get\_min, return\_get\_min, call\_get\_max, return\_get\_max\} \Rightarrow par\_get\_t \in 0..c-1$
inv8: $get\_min\_value\_add \in \{1,2,3\} \lor prog\_basal \in \{return\_get\_min, call\_get\_max, return\_get\_max\} \Rightarrow get\_min\_start\_t = par\_get\_t$
inv10: $get\_max\_value\_add \in \{1,2\} \lor prog\_basal = return\_get\_max \Rightarrow get\_max\_start\_t = par\_get\_t$
inv5:
$(add\_resume = 1 \land prog\_basal = return\_get\_min) \lor add\_resume = 2$
$\Rightarrow$
$((\forall j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \Rightarrow rate\_setting2(j) = -1) \Rightarrow min\_value = c)$
inv6:
$(add\_resume = 1 \land prog\_basal = return\_get\_min) \lor add\_resume = 2$
$\Rightarrow$
$((\exists j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \land rate\_setting2(j) \neq -1) \Rightarrow min\_value = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > par\_get\_t\}))$
inv9:
$prog\_basal = return\_get\_max \Rightarrow$
$max\_value = rate\_setting2(max(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i \leq get\_max\_start\_t\}))$
inv16: $add\_update \in \{1,2\} \Rightarrow par\_get\_t \in dom(rate\_setting2 \rhd \{-1\})$

inv20:
$(add\_start = 1 \land prog\_basal = return\_get\_min) \lor add\_start = 2$
$\Rightarrow$
$((\forall j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \Rightarrow rate\_setting2(j) = -1) \Rightarrow min\_value = c)$

inv19:
$(add\_start = 1 \land prog\_basal = return\_get\_min) \lor add\_start = 2$
$\Rightarrow$
$((\exists j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \land rate\_setting2(j) \neq -1) \Rightarrow min\_value = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > par\_get\_t\}))$

inv22:
$add\_update = 1 \land prog\_basal = return\_get\_min$
$\Rightarrow$
$((\forall j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \Rightarrow rate\_setting2(j) = -1) \Rightarrow min\_value = c)$

inv21:
$add\_update = 1 \land prog\_basal = return\_get\_min$
$\Rightarrow$
$((\exists j \cdot j \in dom(rate\_setting2) \land j > par\_get\_t \land rate\_setting2(j) \neq -1) \Rightarrow min\_value = min(\{i | i \in dom(rate\_setting2 \rhd \{-1\}) \land i > par\_get\_t\}))$

inv23: $add\_resume \in \{1, 2\} \Rightarrow basal\_rate\_in = 0 \land basal\_mode = suspended$

inv24: $add\_update = 1 \Rightarrow basal\_mode = delivering \land prog\_basal \in \{call\_get\_min, return\_get\_min\}$

inv25: $add\_start \in \{1, 2\} \Rightarrow basal\_mode = stop$

EVENTS

Initialisation ⟨extended⟩

begin

act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \,..\, c - 1 \times \{-1\}) \cup \{0 \mapsto 0\}$
act6: $min\_value := 0$
act7: $max\_value := 0$
act11: $get\_min\_value\_add := 0$
act8: $par\_t := 0$
act9: $temp\_min := 0$
act10: $get\_min\_start\_t := 0$
act12: $get\_max\_start\_t := 0$
act13: $get\_max\_value\_add := 0$
act14: $par\_t\_max := 0$
act15: $prog\_basal := null$
act16: $par\_get\_t := 0$
act17: $add\_resume := 0$
act18: $add\_update := 0$
act19: $add\_start := 0$

end

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

extends basal_suspend

when

grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
grd3: $prog\_basal = null$

then

act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$

end

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

refines change_setting

any

t

r

where

grd5: $prog\_basal = null$

grd6: $t \in 0 \mathinner{..} c - 1$

grd7: $rate\_setting2(t) \neq -1$

grd2: $r \in 0 \mathinner{..} basal\_max$

then

act2: $rate\_setting2 := rate\_setting2 \Leftarrow \{t \mapsto r\}$

end

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

refines delete_setting

any

t

where

grd5: $prog\_basal = null$

grd2: $basal\_mode \neq suspended$

grd6: $t \in 1 \mathinner{..} c - 1$

grd7: $rate\_setting2(t) \neq -1$

then

act2: $rate\_setting2 := rate\_setting2 \Leftarrow \{t \mapsto -1\}$

end

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

refines add_setting

any

t

r

where

grd9: $prog\_basal = null$

grd3: $r \in 0 \mathinner{..} basal\_max$

grd4: $basal\_mode \neq suspended$

grd5: $t \in 0 \mathinner{..} c - 1$

grd6: $rate\_setting2(t) = -1$

then

act2: $rate\_setting2 := rate\_setting2 \Leftarrow \{t \mapsto r\}$

end

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

refines basal_resume

when

grd8: $prog\_basal = return\_get\_max$

grd9: $add\_resume = 2$

with

t2: $t2 = min\_value$

t: $t = par\_get\_t$

then

act1: $basal\_rate\_in := max\_value$

act2: $basal\_mode := delivering$

act3: $btime := min\_value - par\_get\_t$

act4: $prog\_basal := null$

act5: $add\_resume := 0$

end

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

any

t

where

grd4: $t \in 0 \mathinner{..} c - 1$

grd5: $prog\_basal = null$

grd6: $add\_resume = 0$

grd1: $basal\_rate\_in = 0$

grd3: $basal\_mode = suspended$

then

act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_resume := 1$

end

Event basal_resume_call_2 ⟨ordinary⟩ $\widehat{=}$

when

grd1: $prog\_basal = return\_get\_min$
grd2: $add\_resume = 1$

then

act1: $prog\_basal := call\_get\_max$
act2: $add\_resume := 2$

end

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

refines rate_update

when

grd12: $add\_update = 1$
grd4: $prog\_basal = return\_get\_min$

with

t2: $t2 = min\_value$
t: $t = par\_get\_t$

then

act1: $basal\_rate\_in := rate\_setting2(par\_get\_t)$
act2: $btime := min\_value - par\_get\_t$
act3: $add\_update := 0$
act4: $prog\_basal := null$

end

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

any

t

where

grd6: $t \in 0 \mathinner{..} c - 1$
grd2: $prog\_basal = null$
grd3: $add\_update = 0$
grd5: $basal\_mode = delivering$
grd7: $rate\_setting2(t) \neq -1$

then

act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_update := 1$

end

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

refines start

when

grd8: $add\_start = 2$
grd9: $prog\_basal = return\_get\_max$

with

t2: $t2 = min\_value$
t: $t = par\_get\_t$

then

act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := max\_value$
act3: $btime := min\_value - par\_get\_t$
act4: $add\_start := 0$
act5: $prog\_basal := null$

end

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

any
t
where
grd1: $t \in 0 \mathinner{..} c - 1$
grd2: $prog\_basal = null$
grd3: $add\_start = 0$
grd4: $basal\_mode = stop$
then
act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_start := 1$
end

Event start_call_2 ⟨ordinary⟩ $\widehat{=}$
when
grd1: $prog\_basal = return\_get\_min$
grd2: $add\_start = 1$
then
act1: $prog\_basal := call\_get\_max$
act2: $add\_start := 2$
end

Event stop ⟨ordinary⟩ $\widehat{=}$
extends stop
when
grd1: $basal\_mode = delivering$
grd2: $prog\_basal = null$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
end

Event get_min_value_1 ⟨ordinary⟩ $\widehat{=}$
refines get_min_value_1
when
grd4: $get\_min\_value\_add = 2$
grd5: $par\_t = c$
then
act1: $min\_value := c$
act2: $get\_min\_value\_add := 0$
act3: $prog\_basal := return\_get\_min$
end

Event get_min_value_2 ⟨ordinary⟩ $\widehat{=}$
refines get_min_value_2
when
grd5: $get\_min\_value\_add = 3$
then
act1: $min\_value := temp\_min$
act2: $get\_min\_value\_add := 0$
act3: $prog\_basal := return\_get\_min$
end

Event get_min_value_start ⟨ordinary⟩ $\widehat{=}$
refines get_min_value_start
when
grd2: $get\_min\_value\_add = 0$
grd3: $prog\_basal = call\_get\_min$
with
t: $t = par\_get\_t$
then
act1: $par\_t := par\_get\_t + 1$
act2: $get\_min\_value\_add := 1$

act3: $get\_min\_start\_t := par\_get\_t$

end

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

refines find_min_value

when

grd1: $par\_t < c$

grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$

grd3: $rate\_setting2(par\_t) = -1$

then

act1: $par\_t := par\_t + 1$

act2: $get\_min\_value\_add := 2$

end

Event find_min_value_2 ⟨ordinary⟩ $\widehat{=}$

refines find_min_value_2

when

grd1: $par\_t < c$

grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$

grd3: $rate\_setting2(par\_t) \neq -1$

then

act1: $temp\_min := par\_t$

act2: $get\_min\_value\_add := 3$

end

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

extends get_max_value

when

grd2: $get\_max\_value\_add = 2$

then

act1: $max\_value := rate\_setting2(par\_t\_max)$

act2: $get\_max\_value\_add := 0$

act3: $prog\_basal := return\_get\_max$

end

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

refines get_max_value_start

when

grd2: $get\_max\_value\_add = 0$

grd3: $prog\_basal = call\_get\_max$

with

t: $t = par\_get\_t$

then

act1: $get\_max\_start\_t := par\_get\_t$

act2: $get\_max\_value\_add := 1$

act3: $par\_t\_max := par\_get\_t$

end

Event get_max_value_1 ⟨ordinary⟩ $\widehat{=}$

extends get_max_value_1

when

grd1: $get\_max\_value\_add = 1$

grd3: $par\_t\_max \geq 0$

grd2: $rate\_setting2(par\_t\_max) = -1$

then

act1: $par\_t\_max := par\_t\_max - 1$

end

Event get_max_value_2 ⟨ordinary⟩ $\widehat{=}$

extends get_max_value_2

when

grd1: $get\_max\_value\_add = 1$

grd2: ⟨theorem⟩ $par\_t\_max \geq 0$

grd3: $rate\_setting2(par\_t\_max) \neq -1$

then

act1: $get\_max\_value\_add := 2$

end

END

MACHINE Basal6_continuous
REFINES Basal6
SEES c_basal2
VARIABLES

basal_rate_in
basal_mode
btime
rate_setting2
min_value
get_min_value_add
par_t
temp_min
get_min_start_t
max_value
get_max_start_t
get_max_value_add
par_t_max
prog_basal
par_get_t
add_resume
add_update
add_start
fbegin
fend
rate_basal_c

INVARIANTS

inv3: $rate\_basal\_c \in \mathbb{N} \pfun 0 \upto basal\_max$
inv4: $fbegin \in dom(rate\_basal\_c)$
inv2: $fend \in 0 \upto c$
inv5: $rate\_basal\_c(fbegin) = basal\_rate\_in$
inv6: $basal\_mode = delivering \limp fend > fbegin$

EVENTS

Initialisation ⟨extended⟩

begin

act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \upto c - 1 \times \{-1\}) \bunion \{0 \mapsto 0\}$
act6: $min\_value := 0$
act7: $max\_value := 0$
act11: $get\_min\_value\_add := 0$
act8: $par\_t := 0$
act9: $temp\_min := 0$
act10: $get\_min\_start\_t := 0$
act12: $get\_max\_start\_t := 0$
act13: $get\_max\_value\_add := 0$
act14: $par\_t\_max := 0$
act15: $prog\_basal := null$
act16: $par\_get\_t := 0$
act17: $add\_resume := 0$
act18: $add\_update := 0$
act19: $add\_start := 0$
act20: $fbegin := 0$
act21: $fend := 0$

act22: $rate\_basal\_c := \{0 \mapsto 0\}$

end

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

extends basal_suspend

any

t

where

grd1: $basal\_rate\_in \neq 0$

grd2: $basal\_mode = delivering$

grd3: $prog\_basal = null$

grd4: $t \in fbegin \mathinner{..} fend$

then

act1: $basal\_rate\_in := 0$

act2: $basal\_mode := suspended$

act3: $rate\_basal\_c := \lambda x \cdot x \geq t | 0$

act4: $fbegin := t$

end

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

extends change_setting

any

t

r

where

grd5: $prog\_basal = null$

grd6: $t \in 0 \mathinner{..} c - 1$

grd7: $rate\_setting2(t) \neq -1$

grd2: $r \in 0 \mathinner{..} basal\_max$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!+ \{t \mapsto r\}$

end

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

extends delete_setting

any

t

where

grd5: $prog\_basal = null$

grd2: $basal\_mode \neq suspended$

grd6: $t \in 1 \mathinner{..} c - 1$

grd7: $rate\_setting2(t) \neq -1$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!+ \{t \mapsto -1\}$

end

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

extends add_setting

any

t

r

where

grd9: $prog\_basal = null$

grd3: $r \in 0 \mathinner{..} basal\_max$

grd4: $basal\_mode \neq suspended$

grd5: $t \in 0 \mathinner{..} c - 1$

grd6: $rate\_setting2(t) = -1$

then

act2: $rate\_setting2 := rate\_setting2 \lhd\!\!\!+ \{t \mapsto r\}$

end

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

extends basal_resume_return

when
grd8: $prog\_basal = return\_get\_max$
grd9: $add\_resume = 2$
then
act1: $basal\_rate\_in := max\_value$
act2: $basal\_mode := delivering$
act3: $btime := min\_value - par\_get\_t$
act4: $prog\_basal := null$
act5: $add\_resume := 0$
act6: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \wedge x < min\_value | max\_value$
act7: $fbegin := par\_get\_t$
act8: $fend := min\_value$
end

Event basal_resume_call ⟨ordinary⟩ $\widehat{=}$
extends basal_resume_call
any
t
where
grd4: $t \in 0 \mathinner{..} c - 1$
grd5: $prog\_basal = null$
grd6: $add\_resume = 0$
grd1: $basal\_rate\_in = 0$
grd3: $basal\_mode = suspended$
then
act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_resume := 1$
end

Event basal_resume_call_2 ⟨ordinary⟩ $\widehat{=}$
extends basal_resume_call_2
when
grd1: $prog\_basal = return\_get\_min$
grd2: $add\_resume = 1$
then
act1: $prog\_basal := call\_get\_max$
act2: $add\_resume := 2$
end

Event rate_update_return ⟨ordinary⟩ $\widehat{=}$
extends rate_update_return
when
grd12: $add\_update = 1$
grd4: $prog\_basal = return\_get\_min$
then
act1: $basal\_rate\_in := rate\_setting2(par\_get\_t)$
act2: $btime := min\_value - par\_get\_t$
act3: $add\_update := 0$
act4: $prog\_basal := null$
act7: $fbegin := par\_get\_t$
act5: $fend := min\_value$
act6: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \wedge x < min\_value | rate\_setting2(par\_get\_t)$
end

Event rate_update_call ⟨ordinary⟩ $\widehat{=}$
extends rate_update_call
any
t
where
grd6: $t \in 0 \mathinner{..} c - 1$
grd2: $prog\_basal = null$

grd3: $add\_update = 0$
grd5: $basal\_mode = delivering$
grd7: $rate\_setting2(t) \neq -1$

then

act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_update := 1$

end

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

when

grd8: $add\_start = 2$
grd9: $prog\_basal = return\_get\_max$

then

act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := max\_value$
act3: $btime := min\_value - par\_get\_t$
act4: $add\_start := 0$
act5: $prog\_basal := null$
act8: $fbegin := par\_get\_t$
act6: $fend := min\_value$
act7: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \wedge x < min\_value | max\_value$

end

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

any

t

where

grd1: $t \in 0 \mathinner{..} c - 1$
grd2: $prog\_basal = null$
grd3: $add\_start = 0$
grd4: $basal\_mode = stop$

then

act1: $par\_get\_t := t$
act2: $prog\_basal := call\_get\_min$
act3: $add\_start := 1$

end

Event start_call_2 ⟨ordinary⟩ $\widehat{=}$
extends start_call_2

when

grd1: $prog\_basal = return\_get\_min$
grd2: $add\_start = 1$

then

act1: $prog\_basal := call\_get\_max$
act2: $add\_start := 2$

end

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

any

t

where

grd1: $basal\_mode = delivering$
grd2: $prog\_basal = null$
grd3: $t \in fbegin \mathinner{..} fend$

then

act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
act3: $fbegin := t$

act4: $rate\_basal\_c := \lambda x \cdot x \geq t | 0$

end

Event get_min_value_1 ⟨ordinary⟩ $\widehat{=}$

extends get_min_value_1

when

grd4: $get\_min\_value\_add = 2$

grd5: $par\_t = c$

then

act1: $min\_value := c$

act2: $get\_min\_value\_add := 0$

act3: $prog\_basal := return\_get\_min$

end

Event get_min_value_2 ⟨ordinary⟩ $\widehat{=}$

extends get_min_value_2

when

grd5: $get\_min\_value\_add = 3$

then

act1: $min\_value := temp\_min$

act2: $get\_min\_value\_add := 0$

act3: $prog\_basal := return\_get\_min$

end

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

extends get_min_value_start

when

grd2: $get\_min\_value\_add = 0$

grd3: $prog\_basal = call\_get\_min$

then

act1: $par\_t := par\_get\_t + 1$

act2: $get\_min\_value\_add := 1$

act3: $get\_min\_start\_t := par\_get\_t$

end

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

extends find_min_value

when

grd1: $par\_t < c$

grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$

grd3: $rate\_setting2(par\_t) = -1$

then

act1: $par\_t := par\_t + 1$

act2: $get\_min\_value\_add := 2$

end

Event find_min_value_2 ⟨ordinary⟩ $\widehat{=}$

extends find_min_value_2

when

grd1: $par\_t < c$

grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$

grd3: $rate\_setting2(par\_t) \neq -1$

then

act1: $temp\_min := par\_t$

act2: $get\_min\_value\_add := 3$

end

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

extends get_max_value

when

grd2: $get\_max\_value\_add = 2$

then

act1: $max\_value := rate\_setting2(par\_t\_max)$

act2: $get\_max\_value\_add := 0$
act3: $prog\_basal := return\_get\_max$

end

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

when

grd2: $get\_max\_value\_add = 0$
grd3: $prog\_basal = call\_get\_max$

then

act1: $get\_max\_start\_t := par\_get\_t$
act2: $get\_max\_value\_add := 1$
act3: $par\_t\_max := par\_get\_t$

end

Event get_max_value_1 ⟨ordinary⟩ $\widehat{=}$
extends get_max_value_1

when

grd1: $get\_max\_value\_add = 1$
grd3: $par\_t\_max \geq 0$
grd2: $rate\_setting2(par\_t\_max) = -1$

then

act1: $par\_t\_max := par\_t\_max - 1$

end

Event get_max_value_2 ⟨ordinary⟩ $\widehat{=}$
extends get_max_value_2

when

grd1: $get\_max\_value\_add = 1$
grd2: ⟨theorem⟩ $par\_t\_max \geq 0$
grd3: $rate\_setting2(par\_t\_max) \neq -1$

then

act1: $get\_max\_value\_add := 2$

end

END

MACHINE Basal6_continuous_2

add a timer

REFINES Basal6_continuous

SEES c_basal2

VARIABLES

basal_rate_in
basal_mode
btime
rate_setting2
min_value
get_min_value_add
par_t
temp_min
get_min_start_t
max_value
get_max_start_t
get_max_value_add
par_t_max
prog_basal
par_get_t
add_resume
add_update
add_start
fbegin
fend
rate_basal_c
time

INVARIANTS

inv1: $time \in 0 \mathinner{..} c - 1$

EVENTS

Initialisation ⟨extended⟩

begin

act2: $basal\_rate\_in := 0$
act3: $basal\_mode := stop$
act4: $btime := c$
act5: $rate\_setting2 := (1 \mathinner{..} c - 1 \times \{-1\}) \cup \{0 \mapsto 0\}$
act6: $min\_value := 0$
act7: $max\_value := 0$
act11: $get\_min\_value\_add := 0$
act8: $par\_t := 0$
act9: $temp\_min := 0$
act10: $get\_min\_start\_t := 0$
act12: $get\_max\_start\_t := 0$
act13: $get\_max\_value\_add := 0$
act14: $par\_t\_max := 0$
act15: $prog\_basal := null$
act16: $par\_get\_t := 0$
act17: $add\_resume := 0$
act18: $add\_update := 0$
act19: $add\_start := 0$
act20: $fbegin := 0$
act21: $fend := 0$
act22: $rate\_basal\_c := \{0 \mapsto 0\}$
act23: $time := 0$

end

Event change_setting ⟨ordinary⟩ $\widehat{=}$
extends change_setting
any
t
r
where
grd5: $prog\_basal = null$
grd6: $t \in 0 \mathinner{..} c - 1$
grd7: $rate\_setting2(t) \neq -1$
grd2: $r \in 0 \mathinner{..} basal\_max$
then
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!+ \{t \mapsto r\}$
end
Event delete_setting ⟨ordinary⟩ $\widehat{=}$
extends delete_setting
any
t
where
grd5: $prog\_basal = null$
grd2: $basal\_mode \neq suspended$
grd6: $t \in 1 \mathinner{..} c - 1$
grd7: $rate\_setting2(t) \neq -1$
then
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!+ \{t \mapsto -1\}$
end
Event add_setting ⟨ordinary⟩ $\widehat{=}$
extends add_setting
any
t
r
where
grd9: $prog\_basal = null$
grd3: $r \in 0 \mathinner{..} basal\_max$
grd4: $basal\_mode \neq suspended$
grd5: $t \in 0 \mathinner{..} c - 1$
grd6: $rate\_setting2(t) = -1$
then
act2: $rate\_setting2 := rate\_setting2 \lhd\!\!+ \{t \mapsto r\}$
end
Event basal_suspend ⟨ordinary⟩ $\widehat{=}$
refines basal_suspend
when
grd1: $basal\_rate\_in \neq 0$
grd2: $basal\_mode = delivering$
grd3: $prog\_basal = null$
grd4: $time \in fbegin \mathinner{..} fend$
with
t: $t = time$
then
act1: $basal\_rate\_in := 0$
act2: $basal\_mode := suspended$
act3: $rate\_basal\_c := \lambda x \cdot x \geq time | 0$
act4: $fbegin := time$
end
Event basal_resume_return ⟨ordinary⟩ $\widehat{=}$
extends basal_resume_return
when
grd8: $prog\_basal = return\_get\_max$

grd9: $add\_resume = 2$

then

act1: $basal\_rate\_in := max\_value$
act2: $basal\_mode := delivering$
act3: $btime := min\_value - par\_get\_t$
act4: $prog\_basal := null$
act5: $add\_resume := 0$
act6: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \land x < min\_value | max\_value$
act7: $fbegin := par\_get\_t$
act8: $fend := min\_value$

end

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

refines basal_resume_call

when

grd5: $prog\_basal = null$
grd6: $add\_resume = 0$
grd1: $basal\_rate\_in = 0$
grd3: $basal\_mode = suspended$

with

t: $t = time$

then

act1: $par\_get\_t := time$
act2: $prog\_basal := call\_get\_min$
act3: $add\_resume := 1$

end

Event basal_resume_call_2 ⟨ordinary⟩ $\widehat{=}$

extends basal_resume_call_2

when

grd1: $prog\_basal = return\_get\_min$
grd2: $add\_resume = 1$

then

act1: $prog\_basal := call\_get\_max$
act2: $add\_resume := 2$

end

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

extends rate_update_return

when

grd12: $add\_update = 1$
grd4: $prog\_basal = return\_get\_min$

then

act1: $basal\_rate\_in := rate\_setting2(par\_get\_t)$
act2: $btime := min\_value - par\_get\_t$
act3: $add\_update := 0$
act4: $prog\_basal := null$
act7: $fbegin := par\_get\_t$
act5: $fend := min\_value$
act6: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \land x < min\_value | rate\_setting2(par\_get\_t)$

end

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

refines rate_update_call

when

grd2: $prog\_basal = null$
grd3: $add\_update = 0$
grd4: $prog\_basal = return\_get\_min$
grd5: $basal\_mode = delivering$
grd7: $rate\_setting2(time) \neq -1$

with

t: $t = time$

then
act1: $par\_get\_t := time$
act2: $prog\_basal := call\_get\_min$
act3: $add\_update := 1$
end
Event start_return ⟨ordinary⟩ $\widehat{=}$
extends start_return
when
grd8: $add\_start = 2$
grd9: $prog\_basal = return\_get\_max$
then
act1: $basal\_mode := delivering$
act2: $basal\_rate\_in := max\_value$
act3: $btime := min\_value - par\_get\_t$
act4: $add\_start := 0$
act5: $prog\_basal := null$
act8: $fbegin := par\_get\_t$
act6: $fend := min\_value$
act7: $rate\_basal\_c := \lambda x \cdot x \geq par\_get\_t \wedge x < min\_value | max\_value$
end
Event start_call ⟨ordinary⟩ $\widehat{=}$
refines start_call
when
grd2: $prog\_basal = null$
grd3: $add\_start = 0$
grd4: $basal\_mode = stop$
with
t: $t = time$
then
act1: $par\_get\_t := time$
act2: $prog\_basal := call\_get\_min$
act3: $add\_start := 1$
end
Event start_call_2 ⟨ordinary⟩ $\widehat{=}$
extends start_call_2
when
grd1: $prog\_basal = return\_get\_min$
grd2: $add\_start = 1$
then
act1: $prog\_basal := call\_get\_max$
act2: $add\_start := 2$
end
Event stop ⟨ordinary⟩ $\widehat{=}$
refines stop
when
grd1: $basal\_mode = delivering$
grd2: $prog\_basal = null$
grd3: $time \in fbegin \mathinner{..} fend$
with
t: $t = time$
then
act1: $basal\_mode := stop$
act2: $basal\_rate\_in := 0$
act3: $fbegin := time$
act4: $rate\_basal\_c := \lambda x \cdot x \geq time | 0$
end
Event get_min_value_1 ⟨ordinary⟩ $\widehat{=}$
extends get_min_value_1

when
grd4: $get\_min\_value\_add = 2$
grd5: $par\_t = c$
then
act1: $min\_value := c$
act2: $get\_min\_value\_add := 0$
act3: $prog\_basal := return\_get\_min$
end
Event get_min_value_2 ⟨ordinary⟩ $\widehat{=}$
extends get_min_value_2
when
grd5: $get\_min\_value\_add = 3$
then
act1: $min\_value := temp\_min$
act2: $get\_min\_value\_add := 0$
act3: $prog\_basal := return\_get\_min$
end
Event get_min_value_start ⟨ordinary⟩ $\widehat{=}$
extends get_min_value_start
when
grd2: $get\_min\_value\_add = 0$
grd3: $prog\_basal = call\_get\_min$
then
act1: $par\_t := par\_get\_t + 1$
act2: $get\_min\_value\_add := 1$
act3: $get\_min\_start\_t := par\_get\_t$
end
Event find_min_value ⟨ordinary⟩ $\widehat{=}$
extends find_min_value
when
grd1: $par\_t < c$
grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$
grd3: $rate\_setting2(par\_t) = -1$
then
act1: $par\_t := par\_t + 1$
act2: $get\_min\_value\_add := 2$
end
Event find_min_value_2 ⟨ordinary⟩ $\widehat{=}$
extends find_min_value_2
when
grd1: $par\_t < c$
grd2: $get\_min\_value\_add = 1 \lor get\_min\_value\_add = 2$
grd3: $rate\_setting2(par\_t) \neq -1$
then
act1: $temp\_min := par\_t$
act2: $get\_min\_value\_add := 3$
end
Event get_max_value ⟨ordinary⟩ $\widehat{=}$
extends get_max_value
when
grd2: $get\_max\_value\_add = 2$
then
act1: $max\_value := rate\_setting2(par\_t\_max)$
act2: $get\_max\_value\_add := 0$
act3: $prog\_basal := return\_get\_max$
end
Event get_max_value_start ⟨ordinary⟩ $\widehat{=}$

extends get_max_value_start

when

grd2: $get\_max\_value\_add = 0$

grd3: $prog\_basal = call\_get\_max$

then

act1: $get\_max\_start\_t := par\_get\_t$

act2: $get\_max\_value\_add := 1$

act3: $par\_t\_max := par\_get\_t$

end

Event get_max_value_1 ⟨ordinary⟩ $\widehat{=}$

extends get_max_value_1

when

grd1: $get\_max\_value\_add = 1$

grd3: $par\_t\_max \geq 0$

grd2: $rate\_setting2(par\_t\_max) = -1$

then

act1: $par\_t\_max := par\_t\_max - 1$

end

Event get_max_value_2 ⟨ordinary⟩ $\widehat{=}$

extends get_max_value_2

when

grd1: $get\_max\_value\_add = 1$

grd2: ⟨theorem⟩ $par\_t\_max \geq 0$

grd3: $rate\_setting2(par\_t\_max) \neq -1$

then

act1: $get\_max\_value\_add := 2$

end

Event timer ⟨ordinary⟩ $\widehat{=}$

when

grd1: $time + 1 \leq c - 1$

grd2:

$\neg($

$(add\_update = 1) \vee$

$((prog\_basal = null) \wedge (add\_update = 0) \wedge (prog\_basal = return\_get\_min) \wedge (basal\_mode = delivering) \wedge (rate\_setting2(time) \neq -1))$

$)$

then

act1: $time := time + 1$

end

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

when

grd1: $time + 1 = c$

grd2:

$\neg($

$(add\_update = 1) \vee$

$((prog\_basal = null) \wedge (add\_update = 0) \wedge (prog\_basal = return\_get\_min) \wedge (basal\_mode = delivering) \wedge (rate\_setting2(time) \neq -1))$

$)$

then

act1: $time := 0$

end

END

MACHINE NormalBolus
SEES c_normalbolus
VARIABLES
insulin_needed
normal_add
normal_delivering_time
normal_delivering_rate
normal_bolus_suspend
INVARIANTS
inv1: $insulin\_needed \in \mathbb{N}$
inv5: $normal\_add \in 0 \mathinner{..} 3$
inv2: $normal\_delivering\_time \in \mathbb{N}$
inv3: $normal\_delivering\_rate \in \mathbb{N}$
inv4: $normal\_delivering\_rate = 0 \lor normal\_delivering\_rate = normal\_bolus\_rate$
inv6: $normal\_add = 0 \Rightarrow normal\_delivering\_rate = 0$
inv7: $normal\_add = 1 \Rightarrow insulin\_needed \neq 0 \land normal\_delivering\_rate = 0$
inv9: $normal\_add = 2 \Rightarrow normal\_delivering\_rate = 0$
inv8: $normal\_add = 3 \Rightarrow normal\_delivering\_rate = normal\_bolus\_rate$
inv10: $normal\_bolus\_suspend \in BOOL$
inv11: $normal\_add = 1 \Rightarrow normal\_bolus\_suspend = FALSE$
inv12: $normal\_add = 2 \Rightarrow normal\_bolus\_suspend = FALSE$
inv19: $normal\_add = 1 \Rightarrow normal\_bolus\_suspend = FALSE \land normal\_delivering\_rate = 0 \land normal\_delivering\_time = 0$
inv20: $normal\_add = 2 \Rightarrow normal\_bolus\_suspend = FALSE \land normal\_delivering\_rate = 0$
inv21: $normal\_add = 3 \Rightarrow normal\_bolus\_suspend = FALSE \land normal\_delivering\_rate > 0$
inv22: $normal\_add = 0 \Rightarrow normal\_delivering\_rate = 0 \land normal\_delivering\_time = 0$
inv23: ⟨theorem⟩ $normal\_bolus\_suspend = TRUE \Rightarrow normal\_add = 0$
EVENTS
Initialisation
begin
act1: $insulin\_needed := 0$
act2: $normal\_delivering\_time := 0$
act3: $normal\_delivering\_rate := 0$
act4: $normal\_add := 0$
act5: $normal\_bolus\_suspend := FALSE$
end
Event normal_bolus_start_calculate_insulin_needed ⟨ordinary⟩ $\widehat{=}$
any
insulin
where
grd1: $insulin > 0$
grd3: $normal\_add = 0$
grd4: $normal\_bolus\_suspend = FALSE$
then
act1: $insulin\_needed := insulin$
act2: $normal\_add := 1$
end
Event normal_bolus_start_calculate_lasting_time ⟨ordinary⟩ $\widehat{=}$
when
grd1: $normal\_add = 1$
then
act1: $normal\_delivering\_time := insulin\_needed / normal\_bolus\_rate$
act2: $insulin\_needed := 0$
act3: $normal\_add := 2$

end

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

when

grd2: $normal\_add = 2$

then

act1: $normal\_delivering\_rate := normal\_bolus\_rate$

act2: $normal\_add := 3$

end

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

when

grd4: $normal\_add = 3$

grd5: $normal\_bolus\_suspend = FALSE$

then

act1: $normal\_delivering\_rate := 0$

act2: $normal\_delivering\_time := 0$

act3: $normal\_add := 0$

act4: $normal\_bolus\_suspend := TRUE$

end

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

when

grd4: $normal\_bolus\_suspend = FALSE$

grd3: $normal\_add = 3$

then

act1: $normal\_delivering\_rate := 0$

act2: $normal\_delivering\_time := 0$

act3: $normal\_add := 0$

end

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

when

grd1: $normal\_bolus\_suspend = TRUE$

grd2: $normal\_add = 0$

then

act1: $normal\_bolus\_suspend := FALSE$

end

END

MACHINE NormalBolus_continuous
REFINES NormalBolus
SEES c_normalbolus_anim
VARIABLES
insulin_needed
normal_add
normal_delivering_time
normal_delivering_rate
normal_delivering_rate_c
nb_now
nb_new_now
normal_bolus_suspend
INVARIANTS
inv1: $normal\_delivering\_rate\_c \in \mathbb{N} \pfun 0 \upto normal\_bolus\_rate$
inv3: $nb\_new\_now \in \mathbb{N}$
inv5: $nb\_now \in dom(normal\_delivering\_rate\_c)$
inv4: $normal\_delivering\_rate\_c(nb\_now) = normal\_delivering\_rate$
EVENTS
Initialisation ⟨extended⟩
begin
act1: $insulin\_needed := 0$
act2: $normal\_delivering\_time := 0$
act3: $normal\_delivering\_rate := 0$
act4: $normal\_add := 0$
act5: $normal\_bolus\_suspend := FALSE$
act8: $normal\_delivering\_rate\_c := \{0 \mapsto 0\}$
act6: $nb\_now := 0$
act7: $nb\_new\_now := 0$
end
Event normal_bolus_start_calculate_insulin_needed ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_start_calculate_insulin_needed
any
$insulin$
where
grd1: $insulin > 0$
grd3: $normal\_add = 0$
grd4: $normal\_bolus\_suspend = FALSE$
then
act1: $insulin\_needed := insulin$
act2: $normal\_add := 1$
end
Event normal_bolus_start_calculate_lasting_time ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_start_calculate_lasting_time
when
grd1: $normal\_add = 1$
then
act1: $normal\_delivering\_time := insulin\_needed / normal\_bolus\_rate$
act2: $insulin\_needed := 0$
act3: $normal\_add := 2$
end
Event normal_bolus_delivery ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_delivery
when
grd2: $normal\_add = 2$
then

act1: $normal\_delivering\_rate := normal\_bolus\_rate$
act2: $normal\_add := 3$
act3: $normal\_delivering\_rate\_c := \lambda t \cdot t \in nb\_now..nb\_now+normal\_delivering\_time | normal\_bolus\_rate$

act5: $nb\_new\_now := nb\_now + normal\_delivering\_time$

end

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

any
ta
where
grd4: $normal\_add = 3$
grd5: $normal\_bolus\_suspend = FALSE$
grd7: $ta \in nb\_now \mathinner{..} nb\_new\_now$
then
act1: $normal\_delivering\_rate := 0$
act2: $normal\_delivering\_time := 0$
act3: $normal\_add := 0$
act4: $normal\_bolus\_suspend := TRUE$
act6: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq ta | 0$
act5: $nb\_now := ta$
end

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

any
ta
where
grd4: $normal\_bolus\_suspend = FALSE$
grd3: $normal\_add = 3$
grd5: $ta = nb\_new\_now$
then
act1: $normal\_delivering\_rate := 0$
act2: $normal\_delivering\_time := 0$
act3: $normal\_add := 0$
act4: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq ta | 0$
act5: $nb\_now := ta$
end

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

any
ta
where
grd1: $normal\_bolus\_suspend = TRUE$
grd2: $normal\_add = 0$
grd3: $ta \geq nb\_now$
then
act1: $normal\_bolus\_suspend := FALSE$
act2: $nb\_now := ta$
act3: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq ta | 0$
end

END

MACHINE NormalBolus_continuous_2
REFINES NormalBolus_continuous
SEES c_normalbolus_anim
VARIABLES
insulin_needed
normal_add
normal_delivering_time
normal_delivering_rate
normal_delivering_rate_c
nb_now
nb_new_now
normal_bolus_suspend
time
t_normal
INVARIANTS
inv1: $time \in \mathbb{N}$
inv2: $t\_normal \in \mathbb{N}$
EVENTS
Initialisation ⟨extended⟩
begin
act1: $insulin\_needed := 0$
act2: $normal\_delivering\_time := 0$
act3: $normal\_delivering\_rate := 0$
act4: $normal\_add := 0$
act5: $normal\_bolus\_suspend := FALSE$
act8: $normal\_delivering\_rate\_c := \{0 \mapsto 0\}$
act6: $nb\_now := 0$
act7: $nb\_new\_now := 0$
act9: $time := 0$
act10: $t\_normal := 0$
end
Event normal_bolus_start_calculate_insulin_needed ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_start_calculate_insulin_needed
any
$insulin$
where
grd1: $insulin > 0$
grd3: $normal\_add = 0$
grd4: $normal\_bolus\_suspend = FALSE$
then
act1: $insulin\_needed := insulin$
act2: $normal\_add := 1$
end
Event normal_bolus_start_calculate_lasting_time ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_start_calculate_lasting_time
when
grd1: $normal\_add = 1$
then
act1: $normal\_delivering\_time := insulin\_needed / normal\_bolus\_rate$
act2: $insulin\_needed := 0$
act3: $normal\_add := 2$
end
Event normal_bolus_delivery ⟨ordinary⟩ $\widehat{=}$
extends normal_bolus_delivery
when

grd2: $normal\_add = 2$

then

act1: $normal\_delivering\_rate := normal\_bolus\_rate$

act2: $normal\_add := 3$

act3: $normal\_delivering\_rate\_c := \lambda t \cdot t \in nb\_now..nb\_now + normal\_delivering\_time | normal\_bolus\_rate$

act5: $nb\_new\_now := nb\_now + normal\_delivering\_time$

act6: $t\_normal := time + normal\_delivering\_time$

end

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

refines normal_bolus_suspend

when

grd4: $normal\_add = 3$

grd5: $normal\_bolus\_suspend = FALSE$

grd7: $time \in nb\_now \,..\, nb\_new\_now$

with

ta: $ta = time$

then

act1: $normal\_delivering\_rate := 0$

act2: $normal\_delivering\_time := 0$

act3: $normal\_add := 0$

act4: $normal\_bolus\_suspend := TRUE$

act6: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq time | 0$

act5: $nb\_now := time$

end

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

refines normal_bolus_finish

when

grd4: $normal\_bolus\_suspend = FALSE$

grd3: $normal\_add = 3$

grd5: $time = nb\_new\_now$

grd6: $t\_normal = time$

with

ta: $ta = time$

then

act1: $normal\_delivering\_rate := 0$

act2: $normal\_delivering\_time := 0$

act3: $normal\_add := 0$

act4: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq time | 0$

act5: $nb\_now := time$

end

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

refines normal_bolus_resume

when

grd1: $normal\_bolus\_suspend = TRUE$

grd2: $normal\_add = 0$

grd3: $time \geq nb\_now$

with

ta: $ta = time$

then

act1: $normal\_bolus\_suspend := FALSE$

act2: $nb\_now := time$

act3: $normal\_delivering\_rate\_c := \lambda t \cdot t \geq time | 0$

end

Event timer ⟨ordinary⟩ $\widehat{=}$

begin

act1: $time := time + 1$

end

END