RTEC unit-tests

Manolis Pitsikalis

June 12, 2018

Introduction

In the following slides, we present:

- A set of fluents used for this set of unit-tests.
- A way to run unit tests in YAP
- A series of test cases along with a short description of each case and the results produced.

testcase/4

Each test case requires its declaration as a *testcase/4* term. The arguments of the *testcase*(*ScenarioId*, *Category*, *TestNumber*, *ExpectedResult*, *ErTimes*) term are defined as follows:

- *ScenarioId* is the name of the scenario to use.
- *Category* is the name of the tests category.
- *TestNumber* is the number of the test for the category it belongs.
- *ExpectedResult* is the expected result of the test.
- *ErTimes* is an argument containing nesessairy information for event recognition. i.e,[(*Step*, *Window*, *StartTime*, *EndTime*),...].

check/3

Similarly with the *testcase/4* term each test case needs a *check*(*Category*, *TestNumber*, *Found*) rule. This user-defined predicate queries the event recognition results and stores them in variable Found in a format matching the corresponding *testscase ExpectedResult*.

Initiation - test 1

Simple narrative.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
lose_wallet(chris)	13	T
win_lottery(chris)	14	I
lose_wallet(chris)	16	T
win_lottery(chris)	18	I
lose_wallet(chris)	21	T

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\ &\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[(10,14),(15,17),(19,22)]
- RTEC = passed
- RTECv2 = passed

Initiation - test 2

Testing simultaneous initiation/termination at $T = 13$ while there is an initiation at $T = 9$.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
lose_wallet(chris)	13	T
win_lottery(chris)	13	I
lose_wallet(chris)	16	T
win_lottery(chris)	18	I
lose_wallet(chris)	21	T

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\ &\text{terminatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

- CurrentTime = 21, WindowSize = 21
- Testing fluent = rich(chris)
- Expected Result=[(10,14),(19,22)]
- RTEC = passed
- RTEC v2 = passed

Initiation - test 3

Testing simultaneous initiation/termination at $T = 13$ while **there isn't** an initiation at $T = 9$.

Table: Event description

Event	Time	Status
lose_wallet(chris)	13	T
win_lottery(chris)	13	I
lose_wallet(chris)	16	T
win_lottery(chris)	18	I
lose_wallet(chris)	21	T

$$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$
$$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[(19,22)]
- RTEC = passed
- RTEC2 = passed

Termination - test 1

Testing termination caused by the initiation of a different fluent using “startI($X = V$)”. “startI($X = V$)” returns the start of interval while “start($X = V$)” returns the start of the interval minus one.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
win_lottery(chris)	13	-
startI(rich(chris)=true)	14	T
go_to(chris,home)	17	T
lose_wallet(chris)	19	-

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[(10,15)]
- RTEC = passed
- RTEC2 = passed

$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$
$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$

$\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).$
$\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\text{startI}(\mathit{rich}(X) = \text{true}), T).$
$\text{terminatedAt}(\mathit{working}(X) = \text{true}, T) \leftarrow$
$\quad \text{happensAt}(\mathit{go\_to}(X, Y), T),$
$\quad Y \backslash = \ \mathit{work}.$

Termination - test 2

Testing termination of fluent A caused by the initiation of a fluent B ($T = 9$, using “startI”), while the event triggering the initiation of B and thus the termination of A, occurs simultaneously with the initiation event of fluent A.

Table: Event description

Event	**Time**	**Status**
go_to(chris,work)	9	I
win_lottery(chris)	9	-
startI(rich(chris)=true)	10	T
go_to(chris,home)	17	T
lose_wallet(chris)	19	-

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[(10,11)]
- RTEC = passed
- RTEC2 = passed

$$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$
$$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$

$$\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).$$
$$\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{startI}(\mathit{rich}(X) = \text{true}), T).$$
$$\text{terminatedAt}(\mathit{working}(X) = \text{true}, T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, Y), T),$$
$$\quad Y \backslash= \mathit{work}.$$

Termination - test 3

Testing termination of a fluent caused by an event that happens simultaneously with its initiation.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
go_to(chris,pub)	9	T
go_to(chris,home)	17	T

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[]
- RTEC = passed
- RTEC v2 = passed

initiatedAt($rich(X)$ = true, T) $\leftarrow$
  happensAt($win\_lottery(X)$, T).
terminatedAt($rich(X)$ = true, T) $\leftarrow$
  happensAt($lose\_wallet(X)$, T).

initiatedAt($working(X)$ = true, T) $\leftarrow$
  happensAt($go\_to(X, work)$, T).
terminatedAt($working(X)$ = true, T) $\leftarrow$
  happensAt(startI($rich(X)$ = true), T).
terminatedAt($working(X)$ = true, T) $\leftarrow$
  happensAt($go\_to(X, Y)$, T),
  $Y \setminus = work$.

Termination - test 4

Testing results when an event terminating a fluent occurs at the same timepoint with the query (Current Time).

Table: Event description

Event	**Time**	**Time**
go_to(chris,work)	9	I
go_to(chris,home)	21	T

$$\begin{array}{l}
\text{initiatedAt}(\textit{working}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{go\_to}(X, \textit{work}), T). \\
\text{terminatedAt}(\textit{working}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\text{startI}(\textit{rich}(X) = \text{true}), T). \\
\text{terminatedAt}(\textit{working}(X) = \text{true}, T) \leftarrow \\
\quad \text{happensAt}(\textit{go\_to}(X, Y), T), \\
\quad Y \setminus = \ \textit{work}.
\end{array}$$

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[(10,22)]
- RTEC = passed
- RTECv2 = passed

termination - test 5

Testing fluent B results, when the initiation event of a fluent A, thus the termination of fluent B, occurs at T=Current Time

Table: event description

Event	Time	Time
go_to(chris,work)	9	$I_{working}$
wins_lottery(chris)	21	I_{rich}
startI(rich(chris))	22	$T_{working}$

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[(10,inf)]
- RTEC = passed
- RTEC v2 = passed

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T). \\
&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).
\end{aligned}
$$

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T). \\
&\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\text{startI}(\mathit{rich}(X) = \text{true}), T). \\
&\text{terminatedAt}(\mathit{working}(X) = \text{true}, T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T), \\
&\quad Y \setminus = \ \mathit{work}.
\end{aligned}
$$

Intervals Manipulation - test 1

Testing interval manipulation predicate “union_all”.

Table: Event description

Event	Time	Status
go_to(chris, work)	9	-
win_lottery(chris)	13	I_{rich}
go_to(chris, pub)	17	$I_{location(chris)=pub}$
lose_wallet(chris)	19	T_{rich}
go_to(chris, home)	21	$T_{location(chris)=pub}$

- Current time = 21, Window = 21
- Testing Fluent = happy(chris)
- Expected Result=[(14,22)]
- RTEC = passed
- RTECv2 = passed

$$\text{initiatedAt}(location(X) = Y,\ T) \leftarrow \\ \quad \text{happensAt}(go\_to(X, Y),\ T).$$

$$\text{initiatedAt}(rich(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(win\_lottery(X),\ T). \\ \text{terminatedAt}(rich(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(lose\_wallet(X),\ T).$$

$$\text{holdsFor}(happy(X) = \text{true},\ I) \leftarrow \\ \quad \text{holdsFor}(rich(X) = \text{true},\ I1) \\ \quad \text{holdsFor}(location(X) = pub,\ I2) \\ \quad \text{union\_all}([I1, I2],\ I).$$

Intervals Manipulation - test 2

Testing interval manipulation predicate “intersect_all”.

Table: Event description

Event	Time	Status
go_to(chris, work)	9	-
win_lottery(chris)	13	I_{rich}
go_to(chris, pub)	17	$I_{location(chris)=pub}$
lose_wallet(chris)	19	T_{rich}
go_to(chris, home)	21	$T_{location(chris)=pub}$

- Current time = 21, Window = 21
- Testing Fluent = infiniteBeers(chris)
- Expected Result=[(18,20)]
- RTEC = passed
- RTECv2 = passed

$$\text{initiatedAt}(location(X) = Y,\ T) \leftarrow$$
$$\quad \text{happensAt}(go\_to(X, Y),\ T).$$

$$\text{initiatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(win\_lottery(X),\ T).$$
$$\text{terminatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(lose\_wallet(X),\ T).$$

$$\text{holdsFor}(infiniteBeers(X) = \text{true},\ I) \leftarrow$$
$$\quad \text{holdsFor}(location(X) = pub,\ I1),$$
$$\quad \text{holdsFor}(rich(X) = \text{true},\ I2),$$
$$\quad \text{intersect\_all}([I1, I2],\ I).$$

Intervals Manipulation - test 3

Testing interval manipulation predicate "relative_complement_all".

Table: Event description

Event	Time	Status
go_to(chris, work)	9	-
win_lottery(chris)	13	I_{rich}
go_to(chris, pub)	17	$I_{location(chris)=pub}$
lose_wallet(chris)	19	T_{rich}
go_to(chris, home)	21	$T_{location(chris)=pub}$

- Current time = 21, Window = 21
- Testing Fluent = shortHappiness(chris)
- Expected Result=[(20,22)]
- RTEC = passed
- RTECv2 = passed

$$\begin{aligned}&\text{initiatedAt}(\mathit{location}(X) = Y,\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{go\_to}(X, Y),\ T).\end{aligned}$$

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

$$\begin{aligned}&\text{holdsFor}(\mathit{shortHappiness}(X) = \text{true},\ I) \leftarrow\\&\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub}, I1),\\&\quad \text{holdsFor}(\mathit{rich}(X) = \text{true}, I2),\\&\quad \text{relative\_complement\_all}(I1, [I2], I).\end{aligned}$$

Intervals Manipulation - test 4

Testing "intersect_all" predicate with a different fluent.

Table: Event description

Event	Time	Status
go_to(chris, work)	9	-
win_lottery(chris)	13	I_{rich}
go_to(chris, pub)	17	$I_{location(chris)=pub}$
lose_wallet(chris)	19	T_{rich}
go_to(chris, home)	21	$T_{location(chris)=pub}$

- Current time = 21, Window = 21
- Testing Fluent = drunk(chris)
- Expected Result=[(18,20)]
- RTEC = passed
- RTECv2 = passed

holdsFor($happy(X)$ = true, I) ←
  holdsFor($rich(X)$ = true, $I1$)
  holdsFor($location(X)$ = pub, $I2$)
  union_all([$I1$, $I2$], I).

holdsFor($infiniteBeers(X)$ = true, I) ←
  holdsFor($location(X)$ = pub, $I1$),
  holdsFor($rich(X)$ = true, $I2$),
  intersect_all([$I1$, $I2$], I).

holdsFor($drunk(X)$ = true, I) ←
  holdsFor($happy(X)$ = true, $I1$),
  holdsFor($infiniteBeers(X)$ = true, $I2$),
  intersect_all([$I1$, $I2$], I).

Intervals Manipulation - test 5

Testing “union_all” predicate with input lists containing more than one interval.

Table: Event description

Event	**Time**	**Status**
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

- Current time = 36, Window = 36
- Testing Fluent = happy(chris)
- Expected Result = [(21,29),(36,inf)]
- RTEC = passed
- RTECv2 = passed

$$\begin{aligned}&\text{initiatedAt}(\mathit{location}(X) = Y,\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{go\_to}(X, Y),\ T).\end{aligned}$$

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

$$\begin{aligned}&\text{holdsFor}(\mathit{happy}(X) = \text{true},\ I) \leftarrow\\&\quad \text{holdsFor}(\mathit{rich}(X) = \text{true},\ I1)\\&\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub},\ I2)\\&\quad \text{union\_all}([I1, I2],\ I).\end{aligned}$$

Intervals Manipulation - test 6

Testing “intersect_all” predicate with input lists containing more than one interval.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

- Current time = 36, Window = 36
- Testing Fluent = infiniteBeers(chris)
- Expected Result = [(22,24),(25,28)]
- RTEC = passed
- RTECv2 = passed

$$\text{initiatedAt}(\mathit{location}(X) = Y,\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, Y),\ T).$$

$$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$
$$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$

$$\text{holdsFor}(\mathit{infiniteBeers}(X) = \text{true},\ I) \leftarrow$$
$$\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub},\ I1),$$
$$\quad \text{holdsFor}(\mathit{rich}(X) = \text{true},\ I2),$$
$$\quad \text{intersect\_all}([I1, I2],\ I).$$

Intervals Manipulation - test 7

Testing “relative_complement_all“ predicate with input lists containing more than one interval.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

- Current time = 36, Window = 36
- Testing Fluent = shortHappiness(chris)
- Expected Result = [(21,22),(24,25),(28,29)]
- RTEC = passed
- RTECv2 = passed

$$\text{initiatedAt}(location(X) = Y,\ T) \leftarrow \\ \text{happensAt}(go\_to(X, Y),\ T).$$

$$\text{initiatedAt}(rich(X) = \text{true},\ T) \leftarrow \\ \text{happensAt}(win\_lottery(X),\ T).$$
$$\text{terminatedAt}(rich(X) = \text{true},\ T) \leftarrow \\ \text{happensAt}(lose\_wallet(X),\ T).$$

$$\text{holdsFor}(shortHappiness(X) = \text{true},\ I) \leftarrow \\ \text{holdsFor}(location(X) = pub, I1), \\ \text{holdsFor}(rich(X) = \text{true}, I2), \\ \text{relative\_complement\_all}(I1, [I2], I).$$

HoldsAt T - test 2

Simple case testing use of holdsAt inside the body of an initiation rule.

Table: Event description

Event	Time	Status
win_lottery(chris)	8	I_{rich}
start(happy(chris))	9	-
sleep_start(chris)	9	$I_{sleeping}$
start(sleeping(chris))	9	$I_{sleepinghappy}$
sleep_end(chris)	15	$T_{sleeping}$
end(sleeping(chris))	15	$T_{sleepinghappy}$

- Current time = 21, Window = 21
- Testing Fluent = sleepingHappy(chris)
- Expected Result = [(10,16)]
- RTEC = passed
- RTECv2 = passed

$\text{initiatedAt}(\textit{sleeping}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{sleep\_start}(X),\ T).$
$\text{terminatedAt}(\textit{sleeping}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{sleep\_end}(X),\ T).$

$\text{initiatedAt}(\textit{rich}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{win\_lottery}(X),\ T),$
$\text{terminatedAt}(\textit{rich}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{lose\_wallet}(X),\ T).$

$\text{holdsFor}(\textit{happy}(X) = \text{true},\ I) \leftarrow$
$\quad \text{holdsFor}(\textit{rich}(X) = \text{true},\ I1)$
$\quad \text{holdsFor}(\textit{location}(X) = \textit{pub},\ I2)$
$\quad \text{union\_all}([I1, I2],\ I).$

$\text{initiatedAt}(\textit{sleepingHappy}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{start}(\textit{sleeping}(X) = \textit{true}),\ T),$
$\quad \text{holdsAt}(\textit{happy}(X) = \textit{true},\ T).$
$\text{terminatedAt}(\textit{sleepingHappy}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{end}(\textit{sleeping}(X) = \textit{true}),\ T).$

HoldsAt T - test 3

Simple case testing use of holdsAt inside the body of an initiation rule with unsorted narrative.

Table: Event description

Event	Time	Status
start(happy(chris))	6	-
sleep_start(chris)	9	$I_{sleeping}$
win_lottery(chris)	5	I_{rich}
start(sleeping(chris))	9	$I_{sleepinghappy}$
sleep_end(chris)	15	$T_{sleeping}$
end(sleeping(chris))	15	$T_{sleepinghappy}$

- Current time = 21, Window = 21
- Testing Fluent = sleepingHappy(chris)
- Expected Result = [(10,16)]
- RTEC = passed
- RTECv2 = passed

$$\begin{array}{l}
\text{initiatedAt}(\textit{sleeping}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{sleep\_start}(X),\ T). \\
\text{terminatedAt}(\textit{sleeping}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{sleep\_end}(X),\ T).
\end{array}$$

$$\begin{array}{l}
\text{initiatedAt}(\textit{rich}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{win\_lottery}(X),\ T), \\
\text{terminatedAt}(\textit{rich}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{lose\_wallet}(X),\ T).
\end{array}$$

$$\begin{array}{l}
\text{holdsFor}(\textit{happy}(X) = \text{true},\ I) \leftarrow \\
\quad \text{holdsFor}(\textit{rich}(X) = \text{true},\ I1) \\
\quad \text{holdsFor}(\textit{location}(X) = \textit{pub},\ I2) \\
\quad \text{union\_all}([I1, I2],\ I).
\end{array}$$

$$\begin{array}{l}
\text{initiatedAt}(\textit{sleepingHappy}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{start}(\textit{sleeping}(X) = \textit{true}),\ T), \\
\quad \text{holdsAt}(\textit{happy}(X) = \textit{true},\ T). \\
\text{terminatedAt}(\textit{sleepingHappy}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\textit{end}(\textit{sleeping}(X) = \textit{true}),\ T).
\end{array}$$

Hierarchy SimpleFluent $\rightarrow$ SDFluent - test 1

Testing hierarchy between a simple fluent and an SDFluent.

Table: Event description

Event	**Time**	**Status**
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\text{initiatedAt}(location(X) = Y,\ T) \leftarrow$$
$$\text{happensAt}(go\_to(X, Y),\ T).$$

$$\text{initiatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\text{happensAt}(win\_lottery(X),\ T),$$
$$\text{not holdsAt}(sleeping(X) = \text{true},\ T).$$
$$\text{terminatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\text{happensAt}(lose\_wallet(X),\ T).$$

$$\text{holdsFor}(happy(X) = \text{true},\ I) \leftarrow$$
$$\text{holdsFor}(rich(X) = \text{true},\ I1)$$
$$\text{holdsFor}(location(X) = pub,\ I2)$$
$$\text{union\_all}([I1, I2],\ I).$$

- Current time = 36, Window = 36
- Testing Fluent = happy(chris)
- Expected Result = [(21,29),(36,inf)]
- RTEC = passed
- RTECv2 = passed

Hierarchy Events → SimpleFluent - test 2

Testing hierarchy between events and a simple fluent.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	-
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
win_lottery(chris)	35	I_{rich}

$$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$

$$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$

- Current time = 36, Window = 36
- Testing Fluent = rich(chris)
- Expected Result = [(22,24),(25,28),(36,inf)]
- RTEC = passed
- RTECv2 = passed

Hierarchy SDFluent → SimpleFluent - test 3

Testing hierarchy between an SDFluent and a simple fluent.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
startI(happy(chris)	21	I_{shappy}
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
end(happy(chris))	28	T_{shappy}
win_lottery(chris)	35	I_{rich}
startI(happy(chris)	36	I_{shappy}

$$\begin{aligned}
&\text{holdsFor}(\mathit{happy}(X)=\text{true},\ I) \leftarrow\\
&\quad \text{holdsFor}(\mathit{rich}(X)=\text{true},\ I1)\\
&\quad \text{holdsFor}(\mathit{location}(X)=\mathit{pub},\ I2)\\
&\quad \text{union\_all}([I1, I2],\ I).
\end{aligned}$$

$$\begin{aligned}
&\text{initiatedAt}(\mathit{shappy}(X)=\text{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\text{startI}(\mathit{happy}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{shappy}(X)=\text{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\text{end}(\mathit{happy}(X) = \mathit{true}),\ T).
\end{aligned}$$

- Current time = 36, Window = 36
- Testing Fluent = shappy(chris)
- Expected Result = [(22,29),(37,inf)]
- RTEC = passed
- RTEC v2 = passed

Hierarchy SDFluent → SDFluent - test 4

Testing hierarchy between an SDFluent and an SDFluent.

Table: Event description

Event	**Time**	**Status**
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$\text{holdsFor}(\mathit{happy}(X) = \text{true},\ I) \leftarrow$
$\quad \text{holdsFor}(\mathit{rich}(X) = \text{true},\ I1)$
$\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub},\ I2)$
$\quad \text{union\_all}([I1, I2],\ I).$

$\text{holdsFor}(\mathit{infiniteBeers}(X) = \text{true},\ I) \leftarrow$
$\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub},\ I1),$
$\quad \text{holdsFor}(\mathit{rich}(X) = \text{true},\ I2),$
$\quad \text{intersect\_all}([I1, I2],\ I).$

$\text{holdsFor}(\mathit{drunk}(X) = \text{true}, I) \leftarrow$
$\quad \text{holdsFor}(\mathit{happy}(X) = \text{true}, I1),$
$\quad \text{holdsFor}(\mathit{infiniteBeers}(X) = \text{true}, I2),$
$\quad \text{intersect\_all}([I1, I2],\ I).$

- Current time = 36, Window = 36
- Testing Fluent = drunk(chris)
- Expected Result = [(22,24),(25,28)]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 1

Testing simple fluent results when performing queries in a large window.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\ &\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

- Step = 9, Window = 36, Time start = 9, Time end = 36
- Testing Fluent = rich(chris)
- Expected Result (per step) = [[],[(22,24),(25,28)],[(22,24),(25,28),(36,inf)]]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 2,10

Table: Event description

Event	Time	Status
go_to(chris,work)	9	-
sleep_start(chris)	13	$I_{sleeping}$
sleep_end(chris)	14	$T_{sleeping}$
go_to(chris,home)	18	-
sleep_start(chris)	28	$I_{sleeping}$
sleep_end(chris)	32	$T_{sleeping}$
go_to(chris, work)	33	-

$$\text{initiatedAt}(\textit{sleeping}(X) = \text{true}, T) \leftarrow \\ \quad \text{happensAt}(\textit{sleep\_start}(X), T).$$
$$\text{terminatedAt}(\textit{sleeping}(X) = \text{true}, T) \leftarrow \\ \quad \text{happensAt}(\textit{sleep\_end}(X), T).$$

Testing Fluent = sleeping(chris)

- Test 2 : Testing simple fluent results when performing queries in a large overlapping window equal to *End time* − *Start time*.
  - Step = 9, Window = 36,Time start = 9, Time end = 36
  - Expected Result (per step) = [[(14,15)],[(14,15)],[(14,15),(29,33)]]
  - RTEC = passed
  - RTECv2 = passed
- Test 10 : Testing simple fluent results when performing queries using overlapping windows where window size is less than *End time* − *Start time*.
  - Step = 9, Window = 18, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(14,15)],[(14,15)],[(29,33)]]
  - RTEC = passed
  - RTECv2 = passed

Windows-ERs - test 3,11

Table: Event description

Event	Time	Status
go_to(chris, work)	9	$I_{working}$
go_to(chris, home)	18	$T_{working}$
go_to(chris, work)	33	$I_{working}$
win_lottery(chris)	35	-
startI(rich(chris))	36	$T_{working}$

initiatedAt($rich(X)$ = true, T) ←
  happensAt($win\_lottery(X)$, T).
terminatedAt($rich(X)$ = true, T) ←
  happensAt($lose\_wallet(X)$, T).

initiatedAt($working(X)$ = true, T) ←
  happensAt($go\_to(X, work)$, T).
terminatedAt($working(X)$ = true, T) ←
  happensAt(startI($rich(X)$ = true), T).
terminatedAt($working(X)$ = true, T) ←
  happensAt($go\_to(X, Y)$, T),
  $Y \backslash = work$.

Testing Fluent = working(chris)

- Test 3 : Testing simple fluent with "startI" in its rule body, recognition results when performing queries in a large overlapping window equal to *End time* − *Start time*
  - Step = 9, Window = 36, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(10,19)],[(10,19)],[(10,19),(34,37)]]
  - RTEC = passed
  - RTEC v2 = passed
- Test 11 : Testing results, when performing queries using overlapping windows where window size is less than *End time* − *Start time*.
  - Step = 9, Window = 18, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(10,19)],[(10,19)],[(34,37)]]
  - RTEC = passed
  - RTEC v2 = passed

Windows-ERs - test 4

Hierarchy (Simple Fluent → SDFluent)/Union test when performing multiple queries in a large window.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\begin{array}{l}\text{initiatedAt}(location(X)=Y,\ T) \leftarrow \\ \quad \text{happensAt}(go\_to(X,Y),\ T).\end{array}$$

$$\begin{array}{l}\text{initiatedAt}(rich(X)=\text{true},\ T) \leftarrow \\ \quad \text{happensAt}(win\_lottery(X),\ T). \\ \text{terminatedAt}(rich(X)=\text{true},\ T) \leftarrow \\ \quad \text{happensAt}(lose\_wallet(X),\ T).\end{array}$$

$$\begin{array}{l}\text{holdsFor}(happy(X)=\text{true},\ I) \leftarrow \\ \quad \text{holdsFor}(rich(X)=\text{true},\ I1) \\ \quad \text{holdsFor}(location(X)=pub,\ I2) \\ \quad \text{union\_all}([I1, I2],\ I).\end{array}$$

- Step = 9, Window = 36, Time start = 9, Time end = 36
- Testing Fluent = happy(chris)
- Expected Result (per step) = [[],[(21,inf)],[(21,29),(36,inf)]]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 5

Hierarchy (SDFluent $\rightarrow$ SDFluent)/Intersection test when performing multiple queries in a large window.

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\begin{aligned}
&\text{holdsFor}(\mathit{happy}(X)=\text{true},\ I) \leftarrow\\
&\quad \text{holdsFor}(\mathit{rich}(X)=\text{true},\ I1)\\
&\quad \text{holdsFor}(\mathit{location}(X)=\mathit{pub},\ I2)\\
&\quad \text{union\_all}([I1, I2],\ I).
\end{aligned}$$

$$\begin{aligned}
&\text{holdsFor}(\mathit{infiniteBeers}(X)=\text{true},\ I) \leftarrow\\
&\quad \text{holdsFor}(\mathit{location}(X)=\mathit{pub},\ I1),\\
&\quad \text{holdsFor}(\mathit{rich}(X)=\text{true},\ I2),\\
&\quad \text{intersect\_all}([I1, I2],\ I).
\end{aligned}$$

- Step = 9, Window = 36, Time start = 9, Time end = 36
- Testing Fluent = infiniteBeers(chris)
- Expected Result (per step) = [[],[(22,24),(25,28)],[(22,24),(25,28)]]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 6

Checking inf update in next query results.

Table: Event description

Event	**Time**	**Status**
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\begin{aligned}&\text{initiatedAt}(\mathit{location}(X)=Y,\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{go\_to}(X,Y),\ T).\end{aligned}$$

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\&\text{terminatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\end{aligned}$$

$$\begin{aligned}&\text{holdsFor}(\mathit{shortHappiness}(X)=\text{true},\ I) \leftarrow\\&\quad \text{holdsFor}(\mathit{location}(X)=\mathit{pub}, \mathit{I1}),\\&\quad \text{holdsFor}(\mathit{rich}(X)=\text{true}, \mathit{I2}),\\&\quad \text{relative\_complement\_all}(\mathit{I1}, [\mathit{I2}], I).\end{aligned}$$

- Step = 9, Window = 36,Time start = 9, Time end = 36
- Testing Fluent = shortHappiness(chris)
- Expected Result (per step) = [[],[(21,22),(24,25),(28,inf)],[(21,22),(24,25),(28,29)]]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 7

Testing results when performing queries with different windows (not overlapping).

Table: Event description

Event	Time	Status
go_to(chris, pub)	20	$I_{location(chris)=pub}$
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
win_lottery(chris)	24	I_{rich}
lose_wallet(chris)	27	T_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}

$$\begin{aligned}
&\text{holdsFor}(\textit{happy}(X)=\text{true},\ I) \leftarrow\\
&\quad \text{holdsFor}(\textit{rich}(X)=\text{true},\ I1)\\
&\quad \text{holdsFor}(\textit{location}(X)=\textit{pub},\ I2)\\
&\quad \text{union\_all}([I1, I2],\ I).
\end{aligned}$$

$$\begin{aligned}
&\text{holdsFor}(\textit{infiniteBeers}(X)=\text{true},\ I) \leftarrow\\
&\quad \text{holdsFor}(\textit{location}(X)=\textit{pub},\ I1),\\
&\quad \text{holdsFor}(\textit{rich}(X)=\text{true},\ I2),\\
&\quad \text{intersect\_all}([I1, I2],\ I).
\end{aligned}$$

$$\begin{aligned}
&\text{holdsFor}(\textit{drunk}(X)=\text{true}, I) \leftarrow\\
&\quad \text{holdsFor}(\textit{happy}(X)=\text{true}, I1),\\
&\quad \text{holdsFor}(\textit{infiniteBeers}(X)=\text{true}, I2),\\
&\quad \text{intersect\_all}([I1, I2],\ I).
\end{aligned}$$

- Step = 9, Window = 9, Time start = 9,Time end = 36
- Testing Fluent = drunk(chris)
- Expected Result (per step) = [[],[(22,24),(25,28)],[(25,28)]]
- RTEC = passed
- RTECv2 = passed

Windows-ERs - test 8,12

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{working}$
sleep_start(chris)	13	$I_{sleeping}$
sleep_end(chris)	14	$T_{sleeping}$
go_to(chris,home)	18	$T_{working}$
go_to(chris, home)	18	$I_{working}$
sleep_start(chris)	28	$I_{sleeping}$
sleep_end(chris)	32	$T_{sleeping}$
go_to(chris, work)	33	-
go_to(chris,work)	33	$I_{working}$

holdsFor(*sleeping_at_work*(*X*) = true, *I*) ←
  holdsFor(*working*(*X*) = true, *I1*),
  holdsFor(*sleeping*(*X*) = true, *I2*),
  intersect_all([*I1*, *I2*], *I*).

Testing Fluent = sleeping_at_work(chris)]

- Test 8 : Testing results when an initiation takes place at the start of window (working T=9) ((9, 18])
  - Step = 9, Window = 9, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(14,15)],[(14,15)],[(14,15)]]
  - RTEC = passed
  - RTECv2 = passed
- Test 12 : Testing forget mechanism with overlapping windows
  - Step = 9, Window = 18, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(14,15)],[(14,15)],[]]
  - RTEC = passed
  - RTECv2 = passed

Windows-ERs - test 9,13

Testing update of recognised intervals with a fluent that is affected by startI predicate without/with overlapping windows.

Event	Time	Status
go_to(chris,work)	9	$I_{working}$
sleep_start(chris)	13	$I_{sleeping}$
sleep_end(chris)	14	$T_{sleeping}$
go_to(chris,home)	18	$T_{working}$
go_to(chris, home)	18	$T_{working}$
go_to(chris, pub)	20	$I_{location(chris)=pub}$
go_to(chris,work)	33	$I_{working}$
win_lottery(chris)	35	$T_{working}$

holdsFor($workingEfficiently(X)=$true, I) ←
holdsFor($working(X)=$true, $I1$),
holdsFor($sleeping_at_work(X)=$true, $I2$),
relative_complement_all($I1$, [$I2$], I).

Testing Fluent = workingEfficiently(chris)

- Test 9
  - Step = 9, Window = 36, Time start = 9, Time end = 36
  - Expected Result (per step) = [[(10,14),(15,19)],[(10,14),(15,19)],[(10,14),(15,19),(34,37)]]
  - RTEC = passed
  - RTEC v2 = passed
- Test 13
  - Step = 9, Window = 18, Time start = 9, Time end = 36
  - Expected Result (per step) = [(10,14),(15,19)],[(10,14),(15,19)],[(15,19),(34,37)]
  - RTEC = passed
  - RTEC v2 = passed

Unsorted narratives - test 1

Testing results when narrative is unsorted.

Table: Event description

Event	**Time**	**Status**
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
lose_wallet(chris)	27	T_{rich}
win_lottery(chris)	24	I_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}
go_to(chris, pub)	20	$I_{location(chris)=pub}$

$$\text{initiatedAt}(location(X) = Y,\ T) \leftarrow$$
$$\text{happensAt}(go\_to(X, Y),\ T).$$

$$\text{initiatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\text{happensAt}(win\_lottery(X),\ T).$$
$$\text{terminatedAt}(rich(X) = \text{true},\ T) \leftarrow$$
$$\text{happensAt}(lose\_wallet(X),\ T).$$

$$\text{holdsFor}(happy(X) = \text{true},\ I) \leftarrow$$
$$\text{holdsFor}(rich(X) = \text{true},\ I1)$$
$$\text{holdsFor}(location(X) = pub,\ I2)$$
$$\text{union\_all}([I1, I2],\ I).$$

- Current time = 36, Window = 36
- Testing Fluent = happy(chris)
- Expected Result = [[(21,29),(36,inf)]]
- RTEC = passed
- RTECv2 = passed

Unsorted narratives - test 2

Testing results when narrative is unsorted, with windows.

Table: Event description

Event	Time	Status
-	$T < 18$	-
win_lottery(chris)	21	I_{rich}
lose_wallet(chris)	23	T_{rich}
lose_wallet(chris)	27	T_{rich}
win_lottery(chris)	24	I_{rich}
go_to(chris, home)	28	$T_{location(chris)=pub}$
win_lottery(chris)	35	I_{rich}
go_to(chris, pub)	20	$I_{location(chris)=pub}$

$$\begin{aligned}
&\text{initiatedAt}(\mathit{location}(X) = Y,\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y),\ T).
\end{aligned}$$

$$\begin{aligned}
&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T), \\
&\quad \text{not holdsAt}(\mathit{sleeping}(X) = \text{true}, T). \\
&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).
\end{aligned}$$

$$\begin{aligned}
&\text{holdsFor}(\mathit{happy}(X) = \text{true},\ I) \leftarrow \\
&\quad \text{holdsFor}(\mathit{rich}(X) = \text{true},\ I1) \\
&\quad \text{holdsFor}(\mathit{location}(X) = \mathit{pub},\ I2) \\
&\quad \text{union\_all}([I1, I2],\ I).
\end{aligned}$$

- Step = 9, Window = 36, Time start = 9, Time end = 36
- Testing Fluent = happy(chris)
- Expected Result (per step) = [[],[(22,24],[(21,29),(36,inf)]]
- RTEC = passed
- RTECv2 = passed

DeadlinesUE - test 1

Testing termination caused by maxDurationUE.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
lose_wallet(chris)	21	T

$$\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$
$$\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$
$$\mathit{maxDurationUE}(\mathit{rich}(X) = \text{true}, 4).$$

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[[(10,14)]]
- RTECv2 = passed

DeadlinesUE - test 2

Testing normal termination.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
lose_wallet(chris)	11	T

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[[(10,12)]]
- RTECv2 = passed

$$\begin{array}{l}\text{initiatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{win\_lottery}(X),\ T). \\ \text{terminatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T). \\ \mathit{maxDurationUE}(\mathit{rich}(X) = \text{true}, 4).\end{array}$$

DeadlinesUE - test 3

Testing deadline extent with initiation.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
win_lottery(chris)	11	I
lose_wallet(chris)	21	T

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ \ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ \ T). \\
&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ \ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ \ T). \\
&\mathit{maxDurationUE}(\mathit{rich}(X) = \text{true}, 4).
\end{aligned}
$$

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[[(10,16)]]
- RTECv2 = passed

DeadlinesUE - test 4

Succesful attempt with simultaneous initiation.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
win_lottery(chris)	13	I,T
lose_wallet(chris)	21	T

$$\begin{array}{l} \text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{win\_lottery}(X),\ T). \\ \text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T). \\ \mathit{maxDurationUE}(\mathit{rich}(X) = \text{true}, 4). \end{array}$$

- Current time = 21, Window = 21
- Testing Fluent = rich(chris)
- Expected Result=[[(10,18)]]
- RTECv2 = passed

DeadlinesUE - test 5

Simple window test.

Table: Event description

Event	Time	Status
win_lottery(chris)	9	I
lose_wallet(chris)	21	T

initiatedAt($rich(X)$ = true, T) ←
  happensAt($win\_lottery(X)$, T).
terminatedAt($rich(X)$ = true, T) ←
  happensAt($lose\_wallet(X)$, T).
$maxDurationUE(rich(X) = \text{true}, 4)$.

- Step = 10, Window = 15, Time start = 0, Time end = 20
- Testing Fluent = rich(chris)
- Expected Result=[[(10,inf)],[(10,14)]]
- RTECv2 = passed

DeadlinesUE - test 6

Testing result when first initiation is outside window.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
win_lottery(chris)	7	I
lose_wallet(chris)	21	T

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T), \\
&\quad \text{not holdsAt}(\mathit{sleeping}(X) = \text{true},\ T). \\
&\text{terminatedAt}(\mathit{rich}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T). \\
&\mathit{maxDurationUE}(\mathit{rich}(X) = \text{true}, 4).
\end{aligned}
$$

- Step = 5, Window = 8, Time start = 0, Time end = 15
- Testing Fluent = rich(chris)
- Expected Result=[[(2,inf)],[(2,inf)],[(2,12)]]
- RTECv2 = passed

DeadlinesUE - test 7

Testing result when first and second initiation are in different windows.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
win_lottery(chris)	7	I
lose_wallet(chris)	21	T

$$\begin{aligned}&\text{initiatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).\\ &\text{terminatedAt}(\mathit{rich}(X)=\text{true},\ T) \leftarrow\\ &\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).\\ &\mathit{maxDurationUE}(\mathit{rich}(X)=\text{true}, 4).\end{aligned}$$

- Step = 3, Window = 3, Time start = 0, Time end = 12
- Testing Fluent = rich(chris)
- Expected Result=[[(2,inf)],[(2,inf)],[(2,inf)],[(2,12)]]
- RTECv2 = passed

DeadlinesUE - test 8

Testing deadline extension.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
win_lottery(chris)	7	I

$$
\begin{array}{l}
\text{initiatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T). \\
\text{terminatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T). \\
\mathit{maxDurationUE}(\mathit{rich2}(X) = \text{true}, 8).
\end{array}
$$

- Step = 3, Window = 15, Time start = 3, Time end = 15
- Testing Fluent = rich2(chris)
- Expected Result=[[(2,inf)],[(2,inf)],[(2,inf)],[(2,16)]]
- RTECv2 = passed

DeadlinesUE - test 9

Normal termination.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
win_lottery(chris)	7	I
lose_wallet(chris)	8	T

$$\begin{array}{l}\text{initiatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{win\_lottery}(X),\ T). \\ \text{terminatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T). \\ \mathit{maxDurationUE}(\mathit{rich2}(X) = \text{true}, 8).\end{array}$$

- Step = 6, Window = 12, Time start = 0, Time end = 12
- Testing Fluent = rich2(chris)
- Expected Result=[[(2,inf)],[(2,9)]]
- RTECv2 = passed

DeadlinesUE - test 10

Normal termination inside deadline, then initiation.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
lose_wallet(chris)	6	T
win_lottery(chris)	7	I

$$\text{initiatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$$
$$\text{terminatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$$
$$\mathit{maxDurationUE}(\mathit{rich2}(X) = \text{true}, 8).$$

- Step = 3, Window = 17, Time start = 2, Time end = 17
- Testing Fluent = rich2(chris)
- Expected Result=[[(2,inf)],[(2,7),(8,inf)],[(2,7),(8,inf)],[(2,7),(8,inf)],[(2,7),(8,16)]]
- RTECv2 = passed

DeadlinesUE - test 11

Testing deadline extension.

Table: Event description

Event	Time	Status
win_lottery(chris)	1	I
win_lottery(chris)	4	I
win_lottery(chris)	7	I

$\text{initiatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{win\_lottery}(X),\ T).$
$\text{terminatedAt}(\mathit{rich2}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{lose\_wallet}(X),\ T).$
$\mathit{maxDurationUE}(\mathit{rich2}(X) = \text{true}, 8).$

- Step = 5, Window = 15, Time start = 0, Time end = 15
- Testing Fluent = rich2(chris)
- Expected Result=[[(2,inf)],[(2,inf)],[(2,16)]]
- RTECv2 = passed

Deadlines - test 1

Testing deadline succesful attempt.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
go_to(chris,home)	21	T

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[[(10,18)]]
- RTECv2 = passed

$$\begin{aligned}
&\text{initiatedAt}(working(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(go\_to(X, work), T). \\
&\text{terminatedAt}(working(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(go\_to(X, Y),\ T), Y \neq work. \\
&maxDuration(working(X) = \text{true}, 8).
\end{aligned}$$

Deadlines - test 2

Testing normal termination in deadlines.

Table: Event description

Event	**Time**	**Status**
go_to(chris,work)	9	I
go_to(chris,home)	14	T

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[[(10,15)]]
- RTECv2 = passed

$$\begin{array}{l}\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T). \\ \text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T). \\ \mathit{maxDuration}(\mathit{working}(X) = \text{true}, \mathit{8}).\end{array}$$

Deadlines - test 3

Second initiation before the deadline.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
go_to(chris,work)	12	I
go_to(chris,home)	21	T

$$\begin{aligned}&\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).\\&\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow\\&\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T).\\&\mathit{maxDuration}(\mathit{working}(X) = \text{true}, 8).\end{aligned}$$

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[[(10,18)]]
- RTECv2 = passed

Deadlines - test 4

Meet deadline with initiation simultaneously.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
go_to(chris,work)	17	I
go_to(chris,home)	21	T

$$\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).$$
$$\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T).$$
$$\mathit{maxDuration}(\mathit{working}(X) = \text{true}, 8).$$

- Current time = 21, Window = 21
- Testing Fluent = working(chris)
- Expected Result=[[(10,18)]]
- RTECv2 = passed

Deadlines - test 5

Sliding window test.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	I
go_to(chris,home)	21	T

$$\begin{array}{l}\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T). \\ \text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\ \quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T). \\ \mathit{maxDuration}(\mathit{working}(X) = \text{true}, 8).\end{array}$$

- Step = 10, Window = 15, Time start = 0, Time end = 20
- Testing Fluent = working(chris)
- Expected Result=[[(10,inf)],[(10,18)]]
- RTECv2 = passed

Deadlines - test 6

Sliding window test, first/second initiation in different windows.

Table: Event description

Event	Time	Status
go_to(chris,work)	2	I
go_to(chris,work)	7	I
go_to(chris,home)	15	T

$$\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).$$
$$\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T).$$
$$\mathit{maxDuration}(\mathit{working}(X) = \text{true}, \mathit{8}).$$

- Step = 4, Window = 4, Time start = 0, Time end = 16
- Testing Fluent = working(chris)
- Expected Result=[[(3,inf)],[(3,inf)],[(3,11)],[]]
- RTECv2 = passed

Deadlines - test 8

Testing deadline termination without extension.

Table: Event description

Event	Time	Status
go_to(chris,work)	1	I
go_to(chris,work)	4	I
go_to(chris,work)	7	I

$\text{initiatedAt}(\textit{working}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{go\_to}(X, \textit{work}), T).$
$\text{terminatedAt}(\textit{working}(X) = \text{true},\ T) \leftarrow$
$\quad \text{happensAt}(\textit{go\_to}(X, Y), Y \neq \textit{work},\ T).$
$\textit{maxDuration}(\textit{working}(X) = \text{true}, 8).$

- Step = 3, Window = 15, Time start = 3, Time end = 15
- Testing Fluent = working(chris)
- Expected Result=[[(2,inf)],[(2,10)],[(2,10)],[(2,10)]]
- RTECv2 = passed

Deadlines - test 9

Testing normal termination.

Table: Event description

Event	**Time**	**Status**
go_to(chris,work)	1	I
go_to(chris,work)	4	I
go_to(chris,work)	6	I
go_to(chris,home)	8	T

$$\begin{aligned}
&\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T). \\
&\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T). \\
&\mathit{maxDuration}(\mathit{working}(X) = \text{true}, \mathit{8}).
\end{aligned}$$

- Step = 6, Window = 12, Time start = 0, Time end = 12
- Testing Fluent = working(chris)
- Expected Result=[[(2,inf)],[(2,9)]]
- RTECv2 = passed

Deadlines - test 10

Testing normal termination before deadline, then initiation.

Table: Event description

Event	Time	Status
go_to(chris,work)	1	I
go_to(chris,work)	4	I
go_to(chris,home)	6	T
go_to(chris,work)	7	I

$$\begin{aligned}
&\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T). \\
&\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow \\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T). \\
&\mathit{maxDuration}(\mathit{working}(X) = \text{true}, \mathit{8}).
\end{aligned}$$

- Step = 3, Window = 17, Time start = 2, Time end = 17
- Testing Fluent = working(chris)
- Expected Result=[[(2,inf)],[(2,7),(8,inf)],[(2,7),(8,inf)],[(2,7),(8,inf)],[(2,7),(8,16)]]
- RTECv2 = passed

Deadlines - test 11

Testing termination caused by deadline.

Table: Event description

Event	Time	Status
go_to(chris,work)	1	I
go_to(chris,work)	4	I
go_to(chris,work)	7	I

$$\text{initiatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T).$$
$$\text{terminatedAt}(\mathit{working}(X) = \text{true},\ T) \leftarrow$$
$$\quad \text{happensAt}(\mathit{go\_to}(X, Y), Y \neq \mathit{work},\ T).$$
$$\mathit{maxDuration}(\mathit{working}(X) = \text{true}, \mathit{8}).$$

- Step = 5, Window = 15, Time start = 0, Time end = 15
- Testing Fluent = working(chris)
- Expected Result=[[(2,inf)],[(2,10)],[(2,10)]]
- RTECv2 = passed

Cycles - test 1

Cycles test where CurrentTime-Window = 0 (simple narrative).

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}
sleep_end(chris)	18	I_{full}

initially($strength(X) = full$).
initiatedAt($strength(X) = tired$, T) $\leftarrow$
  happensAt($go\_to(X, Y)$, T), $Y \neq work$,
  holdsAt($strength(X) = lowering$), T).
initiatedAt($strength(X) = lowering$, T) $\leftarrow$
  happensAt($go\_to(X, work)$, T),
  holdsAt($strength(X) = full$, T).
initiatedAt($strength(X) = full$, T) $\leftarrow$
  happensAt($sleep\_end(X)$, T),
  holdsAt($strength(X) = tired$), T).

- Current time = 21, Window = 21
- Testing Fluent: strength(chris)=full
- Expected Result=[[]]
- RTECv2 = failed ER number: 0 TP: [], FP: [(19,inf)], FN: []

Cycles - test 2

Cycles test where CurrentTime-Window = -1.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}
sleep_end(chris)	18	I_{full}

- Current time = 21, Window = 22
- Testing Fluent: strength(chris)=full
- Expected Result=[[(0,10),(19,inf)]]
- RTECv2 = passed

$$
\begin{aligned}
&\text{initially}(\mathit{strength}(X) = \mathit{full}).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{tired}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T), Y \neq \mathit{work},\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{lowering}), T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{lowering}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{full}, T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{full}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{sleep\_end}(X), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{tired}), T).
\end{aligned}
$$

Cycles - test 4

Cycles test where CurrentTime-Window = -1 (simple narrative).

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}
sleep_end(chris)	18	I_{full}

$$
\begin{aligned}
&\text{initially}(\mathit{strength}(X) = \mathit{full}).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{tired}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T), Y \neq \mathit{work},\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{lowering}), T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{lowering}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{full}, T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{full}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{sleep\_end}(X), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{tired}), T).
\end{aligned}
$$

- Current time = 21, Window = 22
- Testing Fluent: strength(chris)=lowering
- Expected Result=[[(10,15)]]
- RTECv2 = passed

Cycles - test 6

Cycles test where CurrentTime-Window = -1.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}
sleep_end(chris)	18	I_{full}

$$\begin{aligned}
&\text{initially}(\mathit{strength}(X) = \mathit{full}).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{tired},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T),\ Y \neq \mathit{work},\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{lowering}),\ T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{lowering},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{full},\ T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{full},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{sleep\_end}(X), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{tired}),\ T).
\end{aligned}$$

- Current time = 21, Window = 22
- Testing Fluent: strength(chris)=tired
- Expected Result=[[(15,19)]]
- RTECv2 = passed

Cycles - test 7

Window test where fluent initiation is outside the window.

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}
sleep_end(chris)	18	I_{full}

initially($strength(X) = full$).
initiatedAt($strength(X) = tired$, T) $\leftarrow$
  happensAt($go\_to(X, Y)$, T), $Y \neq work$,
  holdsAt($strength(X) = lowering$), T).
initiatedAt($strength(X) = lowering$, T) $\leftarrow$
  happensAt($go\_to(X, work)$, T),
  holdsAt($strength(X) = full$, T).
initiatedAt($strength(X) = full$, T) $\leftarrow$
  happensAt($sleep\_end(X)$, T),
  holdsAt($strength(X) = tired$), T).

- Step = 5, Window = 6, Time start = 0, Time end = 20
- Testing Fluent : strength(chris)=full
- Expected Result=[[(0,inf)],[(0,10)],[],[(19,inf)]],
- RTECv2 = passed

Cycles - test 10

Testing holdsAt of terminating condition when fluent A initiates at T and an initiating event of fluent B that depends on A happens at T (not concurrent initiation-termination).

Table: Event description

Event	Time	Status
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	9	-
sleep_end(chris)	18	-

initially($strength(X) = full$).
initiatedAt($strength(X) = tired, T$) $\leftarrow$
  happensAt($go\_to(X, Y), T$), $Y \neq work$,
  holdsAt($strength(X) = lowering$), T).
initiatedAt($strength(X) = lowering, T$) $\leftarrow$
  happensAt($go\_to(X, work), T$),
  holdsAt($strength(X) = full, T$).
initiatedAt($strength(X) = full, T$) $\leftarrow$
  happensAt($sleep\_end(X), T$),
  holdsAt($strength(X) = tired$), T).

- Current time=21, Window = 22
- Testing Fluent : strength(chris)=lowering
- Expected Result=[[(10,inf)]],
- RTECv2 = passed

Cycles - test 11

Testing cycles results with different fluents.

Table: Event description

Event	**Time**	**Status**
smell_bacon(chris)	1	I_h
found_bacon(chris)	7	I_e
ate_bacon(chris)	10	I_{nFN}, $T_{h,e}$
smell_bacon(chris)	12	-
needs_food(chris)	15	T_{nFN}

initiatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($smell\_bacon(X)$, T),
  not holdsAt($noFoodNeeds(X) = true$), T).
terminatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($found\_bacon(X)$, T),
  holdsAt($hungry(X) = true$), T).
terminatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T),
  holdsAt($eating(X) = true$), T).
terminatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($needsFood(X)$, T)

- Current time=21, Window = 22
- Testing Fluent : hungry(chris)=true
- Expected Result=[[(2,11)]],
- RTECv2 = passed

Cycles - test 12

Testing cycles results with different fluents.

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	7	I_e
ate_bacon(chris)	10	I_{nFN}, $T_{h,e}$
smell_bacon(chris)	12	-
needs_food(chris)	15	T_{nFN}

initiatedAt($hungry(X) = true$, T) $\leftarrow$
 happensAt($smell\_bacon(X)$, T),
 not holdsAt($noFoodNeeds(X) = true$), T).
terminatedAt($hungry(X) = true$, T) $\leftarrow$
 happensAt($ate\_bacon(X)$, T)
initiatedAt($eating(X) = true$, T) $\leftarrow$
 happensAt($found\_bacon(X)$, T),
 holdsAt($hungry(X) = true$), T).
terminatedAt($eating(X) = true$, T) $\leftarrow$
 happensAt($ate\_bacon(X)$, T)
initiatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
 happensAt($ate\_bacon(X)$, T),
 holdsAt($eating(X) = true$), T).
terminatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
 happensAt($needsFood(X)$, T)

- Current time=21, Window = 22
- Testing Fluent : eating(chris)=true
- Expected Result=[[(8,11)]],
- RTECv2 = passed

Cycles - test 13

Testing cycles results with different fluents.

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	7	I_e
ate_bacon(chris)	10	I_{nFN}, $T_{h,e}$
smell_bacon(chris)	12	-
needs_food(chris)	15	T_{nFN}

$$\begin{aligned}
&\text{initiatedAt}(hungry(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(smell\_bacon(X),\ T),\\
&\quad not\ \text{holdsAt}(noFoodNeeds(X) = true),\ T).\\
&\text{terminatedAt}(hungry(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(ate\_bacon(X),\ T)\\
&\text{initiatedAt}(eating(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(found\_bacon(X),\ T),\\
&\quad \text{holdsAt}(hungry(X) = true),\ T).\\
&\text{terminatedAt}(eating(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(ate\_bacon(X),\ T)\\
&\text{initiatedAt}(noFoodNeeds(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(ate\_bacon(X),\ T),\\
&\quad \text{holdsAt}(eating(X) = true),\ T).\\
&\text{terminatedAt}(noFoodNeeds(X) = true,\ T) \leftarrow\\
&\quad \text{happensAt}(needsFood(X),\ T)
\end{aligned}$$

- Current time=21, Window = 22
- Testing Fluent : noFoodNeeds(chris)=true
- Expected Result=[[(11,16)]]
- RTECv2 = passed

Cycles - test 14

Testing cycles results with different fluents - initiation/termination in different windows.

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	7	I_e
ate_bacon(chris)	10	I_{nFN}, $T_{h,e}$
smell_bacon(chris)	12	-
needs_food(chris)	15	T_{nFN}

$$\begin{aligned}
&\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),\\
&\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{needsFood}(X),\ T)
\end{aligned}$$

- Step = 5, Window = 6, Time start = 0, Time end = 15
- Testing Fluent : hungry(chris)=true
- Expected Result=[[(2,inf)],[(2,11)],[(2,11)]]
- RTECv2 = passed

Cycles - test 15

Testing cycles results with different fluents.

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	7	I_e
ate_bacon(chris)	10	I_{nFN}, $T_{h,e}$
smell_bacon(chris)	12	-
needs_food(chris)	15	T_{nFN}

initiatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($smell\_bacon(X)$, T),
  not holdsAt($noFoodNeeds(X) = true$), T).
terminatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($found\_bacon(X)$, T),
  holdsAt($hungry(X) = true$), T).
terminatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T),
  holdsAt($eating(X) = true$), T).
terminatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($needsFood(X)$, T)

- Step = 5, Window = 6, Time start = 0, Time end = 15
- Testing Fluent : eating(chris)=true
- Expected Result=[[],[(8,11)],[(8,11)]]
- RTECv2 = passed

Cycles - test 16

Testing cycles results with concurrent initiation termination.

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
ate_bacon(chris)	1	T_h
smell_bacon(chris)	12	I_h
needs_food(chris)	15	

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),\\
&\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{needsFood}(X),\ T)
\end{aligned}
$$

- Current time=21, Window = 21
- Testing Fluent : hungry(chris)=true
- Expected Result=[[(13,inf)]],
- RTECv2 = passed

Cycles unsorted narrative - Test 1

Cycles test where narrative is unsorted.

Table: Event description

Event	Time	Status
sleep_end(chris)	18	I_{full}
go_to(chris,work)	9	$I_{lowering}$
go_to(chris,home)	14	I_{tired}

- Current time = 21, Window = 22
- Testing Fluent: strength(chris)=full
- Expected Result=[[(0,10),(19,inf)]]
- RTECv2 = passed

$$
\begin{aligned}
&\text{initially}(\mathit{strength}(X) = \mathit{full}).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{tired},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T),\ Y \neq \mathit{work},\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{lowering}), T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{lowering},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{full},\ T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{full},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{sleep\_end}(X), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{tired}), T).
\end{aligned}
$$

Cycles unsorted narrative - Test 2

Cycles test where narrative is unsorted.

Table: Event description

Event	Time	Status
sleep_end(chris)	18	I_{full}
go_to(chris,home)	14	I_{tired}
go_to(chris,work)	9	$I_{lowering}$

$$
\begin{aligned}
&\text{initially}(\mathit{strength}(X) = \mathit{full}).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{tired}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, Y), T), Y \neq \mathit{work},\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{lowering}), T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{lowering}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{go\_to}(X, \mathit{work}), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{full}, T).\\
&\text{initiatedAt}(\mathit{strength}(X) = \mathit{full}, T) \leftarrow\\
&\quad \text{happensAt}(\mathit{sleep\_end}(X), T),\\
&\quad \text{holdsAt}(\mathit{strength}(X) = \mathit{tired}), T).
\end{aligned}
$$

- Step = 10, Window = 21, Time start = 0, Time end = 20
- Testing Fluent: strength(chris)=full
- Expected Result=[[(0,inf)],[(0,10),(19,inf)]]
- RTECv2 = passed

Cycles Deadlines - test 1

Testing cycles-deadlines results (simple test).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_bacon(chris)	10	$T_{h,e}$
ate_food(chris)	10	I_{nFN}
smell_bacon(chris)	12	-

$\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),$
$\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)$
$\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),$
$\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)$
$\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_food}(X),\ T),$
$\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{needsFood}(X),\ T)$
$\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5).$
$\mathit{maxDuration}(\mathit{hungry}(X), 5).$

- Current time=21, Window = 21
- Testing Fluent : hungry(chris)=true
- Expected Result=[[(2,7)]],
- RTECv2 = passed

Cycles Deadlines - test 2

Testing cycles-deadlines results (simple test).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_bacon(chris)	10	$T_{h,e}$
ate_food(chris)	10	I_{nFN}
smell_bacon(chris)	12	-

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),\\
&\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_food}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{needsFood}(X),\ T)\\
&\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5).\\
&\mathit{maxDuration}(\mathit{hungry}(X), 5).
\end{aligned}
$$

- Current time=21, Window = 21
- Testing Fluent : eating(chris)=true
- Expected Result=[[(4,11)]],
- RTECv2 = passed

Cycles Deadlines - test 3

Testing cycles-deadlines results with different fluents (terminated by deadline).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_bacon(chris)	10	$T_{h,e}$
ate_food(chris)	10	I_{nFN}
smell_bacon(chris)	12	-

$\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),$
$\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)$
$\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),$
$\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)$
$\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{ate\_food}(X),\ T),$
$\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).$
$\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow$
$\quad \text{happensAt}(\mathit{needsFood}(X),\ T)$
$\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5).$
$\mathit{maxDuration}(\mathit{hungry}(X), 5).$

- Current time=21, Window = 21
- Testing Fluent : noFoodNeeds(chris)=true
- Expected Result=[[(11,16)]],
- RTECv2 = passed

Cycles Deadlines - test 4

Initiation and termination in different windows (terminated by deadline maxDurationUE) .

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_bacon(chris)	8	$T_{h,e}$
ate_food(chris)	8	I_{nFN}
smell_bacon(chris)	16	I_h

initiatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($smell\_bacon(X)$, T),
  not holdsAt($noFoodNeeds(X) = true$), T).
terminatedAt($hungry(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($found\_bacon(X)$, T),
  holdsAt($hungry(X) = true$), T).
terminatedAt($eating(X) = true$, T) $\leftarrow$
  happensAt($ate\_bacon(X)$, T)
initiatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($ate\_food(X)$, T),
  holdsAt($eating(X) = true$), T).
terminatedAt($noFoodNeeds(X) = true$, T) $\leftarrow$
  happensAt($needsFood(X)$, T)
$maxDurationUE(noFoodNeeds(X), 5)$.
$maxDuration(hungry(X), 5)$.

- Step = 5, Window = 6, Time start = 0, Time end = 20
- Testing Fluent : noFoodNeeds(chris)=true
- Expected Result = [[],[(9,inf)],[(9,14)],[]]
- RTECv2 = passed

Cycles Deadlines - test 5

initiation/termination in different windows (maxDuration).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_bacon(chris)	8	$T_{h,e}$
ate_food(chris)	8	I_{nFN}
smell_bacon(chris)	16	I_h

$$
\begin{array}{l}
\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T), \\
\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T), \\
\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_food}(X),\ T), \\
\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{needsFood}(X),\ T) \\
\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5). \\
\mathit{maxDuration}(\mathit{hungry}(X), 5).
\end{array}
$$

- Step = 5, Window = 6, Time start = 0, Time end = 20
- Testing Fluent : hungry(chris)=true
- Expected Result=[[(2,inf)],[(2,7)],[],[(17,inf)]]
- RTECv2 = passed

Cycles Deadlines - test 6

First and second initiation inside the same window .

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_food(chris)	7	I_{nFN}
ate_food(chris)	8	I_{nFN}

$$\begin{array}{l}
\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T), \\
\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T), \\
\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_food}(X),\ T), \\
\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{needsFood}(X),\ T) \\
\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5). \\
\mathit{maxDuration}(\mathit{hungry}(X), 5).
\end{array}$$

- Step = 3, Window = 4, Time start = 0, Time end = 15
- Testing Fluent : noFoodNeeds(chris)=true
- Expected Result = [[],[],[(8,inf)],[(8,inf)],[(8,14)]]
- RTECv2 = passed

Cycles Deadlines - test 7

First and second initiation in different windows (maxDurationUE).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
found_bacon(chris)	3	I_e
ate_food(chris)	7	I_{nFN}
ate_food(chris)	10	I_{nFN}

$$
\begin{aligned}
&\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T),\\
&\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T)\\
&\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{ate\_food}(X),\ T),\\
&\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T).\\
&\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow\\
&\quad \text{happensAt}(\mathit{needsFood}(X),\ T)\\
&\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X), 5).\\
&\mathit{maxDuration}(\mathit{hungry}(X), 5).
\end{aligned}
$$

- Step = 3, Window = 4, Time start = 0, Time end = 15
- Testing Fluent : noFoodNeeds(chris)=true
- Expected Result = [[],[],[(8,inf)],[(8,inf)],[(8,16)]]
- RTECv2 = passed

Cycles Deadlines - test 8

First and second initiation in different windows (maxDuration).

Table: Event description

Event	Time	Status
smell_bacon(chris)	1	I_h
smell_bacon(chris)	5	I_h

$$
\begin{array}{l}
\text{initiatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{smell\_bacon}(X),\ T), \\
\quad \mathit{not}\ \text{holdsAt}(\mathit{noFoodNeeds}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{hungry}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{found\_bacon}(X),\ T), \\
\quad \text{holdsAt}(\mathit{hungry}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{eating}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_bacon}(X),\ T) \\
\text{initiatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{ate\_food}(X),\ T), \\
\quad \text{holdsAt}(\mathit{eating}(X) = \mathit{true}),\ T). \\
\text{terminatedAt}(\mathit{noFoodNeeds}(X) = \mathit{true},\ T) \leftarrow \\
\quad \text{happensAt}(\mathit{needsFood}(X),\ T) \\
\mathit{maxDurationUE}(\mathit{noFoodNeeds}(X),\ 5). \\
\mathit{maxDuration}(\mathit{hungry}(X),\ 5).
\end{array}
$$

- Step = 3, Window = 4, Time start = 0, Time end = 6
- Testing Fluent : hungry(chris)=true
- Expected Result = [[(2,inf)],[(2,7)]]
- RTECv2 = passed

The End