RTEC2 compiler unit-tests

Manolis Pitsikalis (updated by Alex Artikis)

October 12, 2019

Presentation Structure

- Execution instructions.
- Unit test structure.
- Presentation of the tests used for the evaluation of RTEC2 compiler.

Execution instructions

To run the unit tests of RTEC's compiler, run the `runallcompilertests.sh` script; eg:

```
> ./runallcompilertests.sh
```

Unit test structure

Each unit test includes:

- `rules.prolog`: manually constructed, non-compiled rules.
- `declarations.prolog`: manually constructed declarations.
- `rules_compiled_c.prolog`: rules compiled by means of RTEC's compiler.
- `rules_compiled_t.prolog`: manually compiled rules.

Unit testing then amounts to:

- invoking the compiler to compile `rules.prolog` into `rules_compiled_c.prolog` using the `declarations`;
- comparing `rules_compiled_c.prolog` and `rules_compiled_t.prolog`.

Simple Fluents - test 1

Happens at input events.
User rules:

```
initiatedAt(sleeping(X)=true,T) :-
    happensAt(sleep_start(X),T). %input event
terminatedAt(sleeping(X)=true,T) :-
    happensAt(sleep_end(X),T).   %input event
```

Typed rules:

```
initiatedAt(sleeping(X)=true, T1, T, T2) :-
    happensAtIE(sleep_start(X),T),
    T1=<T,
    T<T2.
terminatedAt(sleeping(X)=true, T1, T, T2) :-
    happensAtIE(sleep_end(X),T),
    T1=<T,
    T<T2.
```

Compiled rules

```
initiatedAt(sleeping(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(sleep_start(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
terminatedAt(sleeping(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(sleep_end(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
```

Status: Passed

Simple Fluents - test 2

HoldsAt of a simple fluent.
User rules:

```
initiatedAt(rich(X)=true, T) :-
    happensAt(win_lottery(X), T),  %input event
    \+holdsAt(sleeping(X)=true,T). %simple fluent
terminatedAt(rich(X)=true, T) :-
    happensAt(lose_wallet(X), T).  %input event
```

Typed rules:

```
initiatedAt(rich(X)=true, T1, T, T2) :-
    happensAtIE(win_lottery(X),T),T1=<T,T<T2,
    \+holdsAtProcessedSimpleFluent(X,sleeping(X)=true,T).
terminatedAt(rich(X)=true, T1, T, T2) :-
    happensAtIE(lose_wallet(X),T),
    T1=<T,
    T<T2.
```

Compiled rules

```
initiatedAt(rich(_131139)=true, _131158, _131124, _131160) :-
     happensAtIE(win_lottery(_131139),_131124),_131158=<_131124,_131124<_131160,
     \+holdsAtProcessedSimpleFluent(_131139,sleeping(_131139)=true,_131124).
terminatedAt(rich(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(lose_wallet(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
```

Status: Passed

Simple Fluents - test 3

Happens at start/end of a SDFluent.
User rules:

```
initiatedAt(shappy(X)=true,T):-
    happensAt(start(happy(X)=true),T). % simple fluent
terminatedAt(shappy(X)=true,T):-
    happensAt(end(happy(X)=true),T).   % simple fluent
```

Typed rules:

```
initiatedAt(shappy(X)=true, T1, T, T2) :-
    happensAtProcessedSDFluent(X,start(happy(X)=true),T),
    T1=<T,
    T<T2.
terminatedAt(shappy(X)=true, T1, T, T2) :-
    happensAtProcessedSDFluent(X,end(happy(X)=true),T),
    T1=<T,
    T<T2.
```

Compiled rules

```
initiatedAt(shappy(_131139)=true, _131150, _131124, _131152) :-
     happensAtProcessedSDFluent(_131139,start(happy(_131139)=true),_131124),
     _131150=<_131124,
     _131124<_131152.
terminatedAt(shappy(_131139)=true, _131150, _131124, _131152) :-
     happensAtProcessedSDFluent(_131139,end(happy(_131139)=true),_131124),
     _131150=<_131124,
     _131124<_131152.
```

Status: Passed

Simple Fluents - test 4

HoldsAt SDFluent.
User rules:

```
initiatedAt(rich(X)=true, T) :-
    happensAt(win_lottery(X), T),  %input event
    \+holdsAt(sleeping_at_work(X)=true,T). %SDfluent
terminatedAt(rich(X)=true, T) :-
    happensAt(lose_wallet(X), T).  %input event
```

Typed rules:

```
initiatedAt(rich(X)=true, T1, T, T2) :-
     happensAtIE(win_lottery(X),T),T1=<T,T<T2,
     \+holdsAtProcessedSDFluent(X,sleeping_at_work(X)=true,T).
terminatedAt(rich(X)=true, T1, T, T2) :-
     happensAtIE(lose_wallet(X),T),
     T1=<T,T<T2.
```

Compiled rules

```
initiatedAt(rich(_131139)=true, _131158, _131124, _131160) :-
     happensAtIE(win_lottery(_131139),_131124),_131158=<_131124,_131124<_131160,
     \+holdsAtProcessedSDFluent(_131139,sleeping_at_work(_131139)=true,_131124).
terminatedAt(rich(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(lose_wallet(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
```

Status: Passed

Simple Fluents - test 5

Happens at start/end simple fluent.
User rules:

```
initiatedAt(srich(X)=true, T) :-
    happensAt(start(rich(X)=true),T).
terminatedAt(srich(X)=true, T) :-
    happensAt(end(rich(X)=true),T).
```

Typed rules:

```
initiatedAt(srich(X)=true, T1, T, T2) :-
    happensAtProcessedSimpleFluent(X,start(rich(X)=true),T),
    T1=<T,T<T2.
terminatedAt(srich(X)=true, T1, T, T2) :-
    happensAtProcessedSimpleFluent(X,end(rich(X)=true),T),
    T1=<T,T<T2.
```

Compiled rules

```
initiatedAt(srich(_131139)=true, _131150, _131124, _131152) :-
    happensAtProcessedSimpleFluent(_131139,start(rich(_131139)=true),_131124),
    _131150=<_131124,
    _131124<_131152.
terminatedAt(srich(_131139)=true, _131150, _131124, _131152) :-
    happensAtProcessedSimpleFluent(_131139,end(rich(_131139)=true),_131124),
    _131150=<_131124,
    _131124<_131152.
```

Status: Passed

SDFluents - test 1

Holds for simple fluent.
User rules:

```
holdsFor(infiniteBeers(X)=true, I) :-
    holdsFor(location(X)=pub, I1), %simple fluent
    holdsFor(rich(X)=true, I2),    %simple fluent
    intersect_all([I1,I2], I).
```

Typed rules:

```
holdsForSDFluent(infiniteBeers(X)=true,I) :-
    holdsForProcessedSimpleFluent(X,location(X)=pub,I1),
    holdsForProcessedSimpleFluent(X,rich(X)=true,I2),
    intersect_all([I1,I2],I).
```

Compiled rules:

```
holdsForSDFluent(infiniteBeers(_131139)=true,_131124) :-
     holdsForProcessedSimpleFluent(_131139,location(_131139)=pub,_131145),
     holdsForProcessedSimpleFluent(_131139,rich(_131139)=true,_131156),
     intersect_all([_131145,_131156],_131124).
```

Status: Passed

SDFluents - test 2

Holds for SDFluent.
User rules:

```
holdsFor(drunk(X)=true,I) :-
    holdsFor(happy(X)=true,I1), %SDFluent
    holdsFor(infiniteBeers(X)=true,I2), %SDFluent
    intersect_all([I1,I2], I).
```

Typed rules:

```
holdsForSDFluent(drunk(X)=true,I) :-
    holdsForProcessedSDFluent(X,happy(X)=true,I1),
    holdsForProcessedSDFluent(X,infiniteBeers(X)=true,I2),
    intersect_all([I1,I2],I).
```

Compiled rules:

```
holdsForSDFluent(drunk(_131139)=true,_131124) :-
     holdsForProcessedSDFluent(_131139,happy(_131139)=true,_131145),
     holdsForProcessedSDFluent(_131139,infiniteBeers(_131139)=true,_131156),
     intersect_all([_131145,_131156],_131124).
```

Status: Passed

Cycles - test 1

Holds at cyclic.
Typed rules:

```
initiatedAt(strength(X)=full, T1, -1, T2) :-
    T1=< -1,-1<T2.
initiatedAt(strength(X)=tired, T1, T, T2) :-
    happensAtIE(ends_working(X),T),
    T1=<T,T<T2,
    holdsAtCyclic(X,strength(X)=lowering,T).
initiatedAt(strength(X)=lowering, T1, T, T2) :-
    happensAtIE(starts_working(X),T),
    T1=<T,T<T2,
    holdsAtCyclic(X,strength(X)=full,T).
initiatedAt(strength(X)=full, T1, T, T2) :-
    happensAtIE(sleep_end(X),T),
    T1=<T,T<T2,
    holdsAtCyclic(X,strength(X)=tired,T).
```

Compiled rules:

```
initiatedAt(strength(_131143)=full, _131124, -1, _131126) :-
     _131124=< -1,
    -1<_131126.
initiatedAt(strength(_131139)=tired, _131159, _131124, _131161) :-
     happensAtIE(ends_working(_131139),_131124),_131159=<_131124,_131124<_131161,
     holdsAtCyclic(_131139,strength(_131139)=lowering,_131124).
initiatedAt(strength(_131139)=lowering, _131159, _131124, _131161) :-
     happensAtIE(starts_working(_131139),_131124),_131159=<_131124,_131124<_131161,
     holdsAtCyclic(_131139,strength(_131139)=full,_131124).
initiatedAt(strength(_131139)=full, _131159, _131124, _131161) :-
     happensAtIE(sleep_end(_131139),_131124),_131159=<_131124,_131124<_131161,
     holdsAtCyclic(_131139,strength(_131139)=tired,_131124).
```

Status: Passed

MaxDuration/UE - test 1

MaxDuration.
Typed rules:

```
initiatedAt(working(X)=true, T1, T, T2) :-
    happensAtIE(starts_working(X),T),
    T1=<T,
    T<T2.
terminatedAt(working(X)=true, T1, T, T2) :-
    happensAtIE(ends_working(X),T),
    T1=<T,
    T<T2.
maxDuration(working(X)=true,working(X)=false,8) :- grounding(working(X)=true).
```

Compiled rules:

```
initiatedAt(working(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(starts_working(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
terminatedAt(working(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(ends_working(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
maxDuration(working(_131166)=true,working(_131166)=false,8) :-
     grounding(working(_131166)=true).
```

Status: Passed

MaxDuration/UE - test 2

MaxDurationUE.
Typed rules:

```
initiatedAt(rich(X)=true, T1, T, T2) :-
    happensAtIE(win_lottery(X),T),
    T1=<T,
    T<T2.
terminatedAt(rich(X)=true, T1, T, T2) :-
    happensAtIE(lose_wallet(X),T),
    T1=<T,
    T<T2.
maxDurationUE(rich(X)=true,rich(X)=false,4) :- grounding(rich(X)=true).
```

Compiled rules:

```
initiatedAt(rich(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(win_lottery(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
terminatedAt(rich(_131139)=true, _131145, _131124, _131147) :-
     happensAtIE(lose_wallet(_131139),_131124),
     _131145=<_131124,
     _131124<_131147.
maxDurationUE(rich(_131166)=true,rich(_131166)=false,4) :-
     grounding(rich(_131166)=true).
```

Status: Passed

Findall - test 1

Findall on intervals.
User rules:

```
holdsFor(workingEfficiently(X)=true,I):-
    holdsFor(working(X)=true,I1),
    holdsFor(sleeping_at_work(X)=true,I2),
    relative_complement_all(I1,[I2],Ii),
    findall((S,E),(member(Ii,(S,E)),Diff is E - S,compare(>,Diff,2)),I).
```

Typed rules:

```
holdsForSDFluent(workingEfficiently(X)=true,I) :-
     holdsForProcessedSimpleFluent(X,working(X)=true,Iw),
     holdsForProcessedSDFluent(X,sleeping_at_work(X)=true,Isw),
     relative_complement_all(Iw,[Isw],Ii),
     findall((S,E),(member(Ii,(S,E)),Diff is E-S,compare(>,Diff,2)),I).
```

Compiled rules:

```
holdsForSDFluent(workingEfficiently(_131139)=true,_131124) :-
     holdsForProcessedSimpleFluent(_131139,working(_131139)=true,_131145),
     holdsForProcessedSDFluent(_131139,sleeping_at_work(_131139)=true,_131156),
     relative_complement_all(_131145,[_131156],_131168),
     findall((_131176,_131177),(member(_131168,(_131176,_131177)),
                                _131194 is _131177-_131176,
                                compare(>,_131194,2)),
             _131124).
```

Status: Passed

Findall - test 2

Findall on holdsAt.
User rules:

```
holdsFor(workingEfficientlyAtWork(X)=true,I):-
    holdsFor(working(X)=true,I1),
    holdsFor(sleeping_at_work(X)=true,I2),
    relative_complement_all(I1,[I2],Ii),
    findall((S,E),(
                    member(Ii,(S,E)),
                    holdsAt(location(X)=work,S)
                  ),
        I).
```

Typed rules:

```
holdsForSDFluent(workingEfficientlyAtWork(X)=true,I) :-
     holdsForProcessedSimpleFluent(X,working(X)=true,Iw),
     holdsForProcessedSDFluent(X,sleeping_at_work(X)=true,Isw),
     relative_complement_all(Iw,[Isw],Ii),
     findall((S,E),(
                    member(Ii,(S,E)),
                    holdsAtProcessedSimpleFluent(X,location(X)=work,S)
                   )
            ,I).
```

Compiled rules:

```
holdsForSDFluent(workingEfficientlyAtWork(_131139)=true,_131124) :-
     holdsForProcessedSimpleFluent(_131139,working(_131139)=true,_131145),
     holdsForProcessedSDFluent(_131139,sleeping_at_work(_131139)=true,_131156),
     relative_complement_all(_131145,[_131156],_131168),
     findall((_131176,_131177),(
                                 member(_131168,(_131176,_131177)),
                                 holdsAtProcessedSimpleFluent(_131139,location(_131139)=work,_131176)
                                 )
            ,_131124).
```

Status: Passed

The End