Temporal Logic & Rosetta

Kylie Williams

Vision Statement

- Automatic verification in Rosetta:
  - Develop a state space from a facet... Completed.
  - Express properties:
    - Some model checking tools specify properties and the model in different formalisms.
    - Since Rosetta is heterogeneous it would be desirable to be able to specify both properties and the model in Rosetta.
  - Combine the state space and properties and undertake some model checking.
Overview of Temporal Logic

- Temporal logic is a form of logic specifically tailored for statements and reasoning which involve the notion of order in time.
- First suggested by A. Pnueli in 1977
- PTL & CTL*
Path Operators

- $A - A P$ states that all executions out of the current state satisfy $P$.
- $E - E P$ states that from the current state, there exists an execution satisfying $P$.
- The path operators ($A$ & $E$) and temporal operators ($G$ & $F$) are often used in pairs.

Example

- State 0: Warm, Ok
- State 1: Ok
- State 2: Error
Some Properties

- From 0 we have F E X error
- G(warm => F ¬warm)

Rosetta Domain Hierarchy
Addition of Temporal Logic

- State-based
  - PLTL
  -CTL*
- Signal-based
domain RTL :: filters
  execution :: sequence(int)
  property :: <x> (fail :: unit) :: boolean;

X_1 :: <x> (fail :: property :: property ::);
F_1 :: <x> (fail :: property :: property ::);
G_1 :: <x> (fail :: property :: property ::);

satisfied_by::(sigma :: execution :: property :: natural; p :: property :: boolean;

__1

__2

null domain::temporal_logic

Temporal Logic & Rosetta
Alarm Clock

```plaintext
declare alarmClock : time, displayTime, time;
alarmOn : boolean, setAlarm, setTime, alarmToggles : boolean;
begin
  time := currentTime;
  alarm := true;
  if done then
    done := false;
    alarm := false;
  end
end
```

Properties

```plaintext
declare properties(): PLTL is
  alarmOn, alarm, setAlarm, setTime, alarmToggles : boolean;
  displayTime : time;
  x : time;
begin
  reference_RTL : alarmClock(timeIn, displayTime, alarm, setAlarm, setTime, alarmToggles);
  property_one : G(reference_RTL, alarmOn) implies F reference_RTL.alarmOn;
  property_two : G(reference_RTL.alarmOn and reference_RTL.alarmToggles) implies F not reference_RTL.alarmOn;
  property_three : G(reference_RTL.setAlarm and reference_RTL.alarmToggles) implies F reference_RTL.displayTime = x;
  property_four : G(reference_RTL.ctime and reference_RTL.displayTime) implies F reference_RTL.displayTime = x;
end
```
Conclusion

- Still need to consider
  - Fairness properties
  - Weak Until
  - CTL* Domain
  - Temporal logic as applied to signal-based domains.