

DIT635 - Finite State Verification Activity

Temporal Operators: A quick reference list. p is a Boolean predicate or atomic variable.

- $G p$: p holds globally at every state on the path from now until the end
- $F p$: p holds at some future state on the path (but not all future states)
- $X p$: p holds at the next state on the path
- $p U q$: q holds at some state on the path and p holds at every state before the first state at which q holds.
- A : for all paths reaching out from a state, used in CTL as a modifier for the above properties ($AG p$)
- E : for one or more paths reaching out from a state (but not all), used in CTL as a modifier for the above properties ($EF p$)

An LTL example:

- $G (\text{SEND} \rightarrow F (\text{RECEIVED}))$
- It is always true (G), that if property SEND is true, then at some point in the future (F), property RECEIVED will become true.

A CTL example:

- $EG (\text{WIND} \rightarrow AF (\text{RAIN}))$
- There is a potential future where it is a certainty (EG) that - if there is wind (property WIND is true) - it will always be followed eventually (AF) by rain (RAIN is true).

Consider a finite state model of a traffic-light controller for a single direction with a pedestrian crossing and a button to request right-of-way to cross the road.

State variables:

- **traffic_light**: {RED, YELLOW, GREEN}
- **pedestrian_light**: {WAIT, WALK, FLASH}
- **request_button**: {RESET, SET}

Initially, the state is: **traffic_light = RED, pedestrian_light = WAIT, request_button = RESET**

Transitions:

pedestrian_light:

- **WAIT** → **WALK** if **traffic_light = RED**
- **WAIT** → **WAIT** otherwise
- **WALK** → {**WALK, FLASH**}
- **FLASH** → {**FLASH, WAIT**}

traffic_light:

- RED → GREEN if button = RESET
- RED → RED otherwise
- GREEN → {GREEN, YELLOW} if button = SET
- GREEN → GREEN otherwise
- YELLOW → {YELLOW, RED}

reset_button:

- SET → RESET if pedestrian_light = WALK
- SET → SET otherwise
- RESET → {RESET, SET} if traffic_light = GREEN
- RESET → RESET otherwise

1. Briefly describe a safety-property (nothing “bad” ever happens) for this model and formulate it in CTL.

2. Briefly describe a liveness-property (something “good” eventually happens) for this model and formulate it in LTL.