Model Checking: Examples in LTL

Assaf Kfoury

March 15, 2016
top-view of model checking
(using a temporal logic such as LTL, but not only)

▶ what we are given:
1. a transition system $S$, which specifies a protocol for the simultaneous operation – asynchronous or synchronous – of communicating/interacting processes
2. temporal WFF $\varphi$ expressing some property of $S$

▶ what we want to check:
1. do all executions specified by $S$ satisfy $\varphi$?
2. if we cannot answer preceding question, can we determine whether a “significant” subset of all executions specified by $S$ satisfy $\varphi$?
3. preferably in a fully automated way
common properties expressible in LTL

- **safety**
  - “something bad will not happen”
    - $G \neg(\text{reactor\_temp} > 1000)$
    - $G \neg((x = 0) \land X(y = z/x))$
    - $G \neg(\text{system\_crash})$ (the system should never crash)
    - generally $G \neg(\cdots)$

- **liveness**
  - “something good will happen”
    - $G\ (\text{start} \rightarrow F\ \text{terminate})$
    - $G\ (\text{switch\_on} \rightarrow F\ \text{start})$
    - $G\ (\text{switch\_on} \rightarrow X\ \text{start})$ (perhaps too stringent?)
    - $G\ (\text{packet\_sent} \rightarrow F\ \text{packet\_received})$
    - typically $G\ (\cdots \rightarrow F(\cdots))$ or $G\ (\cdots \rightarrow X(\cdots))$
common properties expressible in LTL (continued)

- safety or liveness? 
sometimes both

- “from any state, it is possible to return to a reset state”
  \[ G(\neg\text{reset} \rightarrow F\text{reset}) \]

- “grant a request 3 cycles after receiving the request”
  \[ G(\text{request} \rightarrow X\times X\times\text{grant}) \]
fairness

“If something is attempted/requested infinitely often, then it will be successful/allocated infinitely often”

- $\mathbf{G} \mathbf{F} \text{ready} \rightarrow \mathbf{G} \mathbf{F} \text{run}$
- $\mathbf{G} \mathbf{F} \text{give\_one} \rightarrow \mathbf{G} \mathbf{F} \text{receive\_one}$

- typically $\mathbf{G} \mathbf{F} ( \cdots ) \rightarrow \mathbf{G} \mathbf{F} ( \cdots )$
- fairness w.r.t. a particular $\varphi$, the WFF $\mathbf{G} \mathbf{F} \varphi$ means “$\varphi$ holds infinitely often, if the path is infinite”
  “$\varphi$ holds at the last state, if the path is finite”

(On the next slide fairness is called strong fairness)
finer examination of fairness:
consider many interacting processes, \( i = 1, 2, 3, \ldots \), with
\( \text{en}_i = \text{“}i \text{ is enabled”} \) and \( \text{c}_i = \text{“}i \text{ is executing critical section”} \)

- **absolute fairness**
  for every \( i = 1, 2, \ldots \), expressed as \( \text{GF} \ c_i \)
  but which ignores that \( i \) may not be ready to execute at certain times

- **strong fairness**
  for every \( i = 1, 2, \ldots \), expressed as \( \text{GF} \ \text{en}_i \rightarrow \text{GF} \ (\text{en}_i \land \text{c}_i) \)
  \( \text{i.e., “}i\text{ enabled infinitely often executes crit sect infinitely often”} \)

- **weak fairness**
  for every \( i = 1, 2, \ldots \), expressed as \( \text{FG} \ \text{en}_i \rightarrow \text{GF} \ (\text{en}_i \land \text{c}_i) \)
  \( \text{i.e., “}i\text{ enabled almost always executes crit sect infinitely often”} \)
  (Note that \( \text{GF}(\text{en}_i \land \text{c}_i) \) can be replaced by \( \text{GF} \ c_i \) in weak fairness)
FACT:

- absolute fairness implies weak fairness
- strong fairness implies weak fairness
- absolute fairness does not imply strong fairness
- strong fairness does not imply absolute fairness
common properties expressible in LTL (continued)

- **reachability**
  
  “a particular state is reached from the present state”

  (sometimes treated as a case of **safety**, more on reachability later)

- **deadlock freedom**
  
  “a deadend state will never be reached”

  (sometimes treated as a case of **liveness**, more on deadlocks later)

- **mutual exclusion**
  
  “two processes are not allowed to enter same critical section”

  (sometimes treated as a case of **safety**)

  \[ G \neg (P1_{\text{in\_critical\_section}} \land P2_{\text{in\_critical\_section}}) \]
specific properties, some related to reachability

- “ϕ never holds in two consecutive states”
  \[ \mathsf{G} (\varphi \rightarrow \mathsf{X} \neg \varphi) \]

- “if ϕ holds in state s, then ϕ holds in all states after s”
  \[ \mathsf{G} (\varphi \rightarrow \mathsf{G} \varphi) \]

  why is this different from \[ \mathsf{G} (\varphi \rightarrow \mathsf{F} \varphi) \] ??

- “ϕ holds in at most one state”
  \[ \mathsf{G} (\varphi \rightarrow \mathsf{X} \mathsf{G} \neg \varphi) \]

- “ϕ holds in at least two states”
  \[ \mathsf{F} (\varphi \wedge \mathsf{X} \mathsf{F} \varphi) \]

already seen: “ϕ holds infinitely often”
\[ \mathsf{G} \mathsf{F} \varphi \]

already seen: “eventually ϕ always holds”
\[ \mathsf{F} \mathsf{G} \varphi \]

“unless s is the first state of the path, if ϕ holds in state s, then ϕ must hold in at least one of the two states just before \ s”
\[ (\mathsf{X} \varphi \rightarrow \varphi) \land \mathsf{G} (\mathsf{X} \mathsf{X} \varphi \rightarrow \varphi \lor \mathsf{X} \varphi) \]
specific properties related to **deadlocks**

- “there is no next state”
  \[ X \perp \]

- “every state which has no next state is a **terminal** state”
  \[ G (X \perp \rightarrow \text{terminal}) \]

- “the system is free of deadlocks”
  this is the same as preceding assertion, *i.e.*, \[ G (X \perp \rightarrow \text{terminal}) \]

- “a deadlock state can be reached” (negation of preceding assertion)
  \[ F (X \perp \land \neg \text{terminal}) \]

- “every execution path is finite (system has no infinite execution)”
  \[ F X \perp \]

- “every execution path is infinite (system has no finite execution)”
  \[ G X \top \]
specific properties related to **alternation**

- “\( \varphi \) holds in every odd state and does not hold in every even state”
  
  (states are counted from 1)

  \[ \varphi \land G (\varphi \leftrightarrow X \neg \varphi) \]

- what does the following say:
  
  \[ (\varphi \land G (\varphi \leftrightarrow X \neg \varphi)) \lor X (\varphi \land G (\varphi \leftrightarrow X \neg \varphi)) \]

- how about instead: \( G (\varphi \leftrightarrow X \neg \varphi) \)?

  it is more restrictive than the preceding WFF, as it is satisfied by the *first* and the *second*, but not the *third*, of the following paths:

  - Co: \( \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \cdots \) (\( \varphi \) true in every odd state)
  
  - C: \( \neg \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \cdots \) (\( \varphi \) true in every even state)
  
  - Co: \( \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \cdots \) (\( \varphi \) true in every odd state and in first state)
specific properties related to **alternation** (continued)

- how about the following:
  \[(\varphi \land G (\varphi \leftrightarrow X \neg \varphi)) \land X (\varphi \land G (\varphi \leftrightarrow X \neg \varphi)) \text{ ???}\]
  (contradictory WFF, *i.e.*, complicated way of asserting \(\bot\))
specific properties related to **alternation** (continued)

- “\( \varphi \) holds in every odd state”, *i.e.*, we want:

  \[
  \varphi \rightarrow \text{??} \rightarrow \varphi \rightarrow \text{??} \rightarrow \varphi \rightarrow \text{??} \rightarrow \cdots
  \]

- how about \( \varphi \land G (\varphi \rightarrow XX \varphi) \)?

  a good candidate, but NOT quite, because it is **not** satisfied by a path of the form

  \[
  \varphi \rightarrow \varphi \rightarrow \varphi \rightarrow \neg \varphi \rightarrow \varphi \rightarrow \text{??} \rightarrow \cdots
  \]

- in fact, “\( \varphi \) holds in every odd state” is NOT expressible in LTL
- describe in English the paths satisfying \( G (\varphi \rightarrow XX \varphi) \)
- describe in English the paths satisfying \( \varphi \land G (\varphi \rightarrow XX \varphi) \)
specific properties related to responsiveness

- "every request is eventually acknowledged"
  \[ G (\text{request} \rightarrow \mathbf{X} \mathbf{F} \text{ack}) \]

- "every request remains true until it is acknowledged"
  \[ G (\text{request} \rightarrow (\text{request} \mathbf{U} \text{ack})) \]

- "every request remains true until it is acknowledged, after which it immediately becomes false"
  \[ G (\text{request} \rightarrow ((\text{request} \land \neg \text{ack}) \mathbf{U} (\neg \text{request} \land \text{ack}))) \]