Model Checking: Examples in LTL

Assaf Kfoury

March 19, 2015 (modified March 23, 2015)
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”
  - \( \mathbf{G} \neg(\text{reactor\_temp} > 1000) \)
  - \( \mathbf{G} \neg((x = 0) \land \mathbf{X}(y = z/x)) \)
  - \( \mathbf{G} \neg(\text{system\_crash}) \) (the system should never crash)
  - generally \( \mathbf{G} \neg(\cdots) \)

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

- **fairness**
  
  “if something is attempted/requested infinitely often, then it will be successful/allocated infinitely often”

  - $\mathbf{G} \mathbf{F}$ ready $\rightarrow \mathbf{G} \mathbf{F}$ run
  - $\mathbf{G} \mathbf{F}$ give_one $\rightarrow \mathbf{G} \mathbf{F}$ receive_one

  - typically $\mathbf{G} \mathbf{F}$ (· · ·) $\rightarrow \mathbf{G} \mathbf{F}$ (· · ·)
  - 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”
common properties expressible in LTL (continued)

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

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

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

- **weak fairness**
  for every \( i = 1, 2, \ldots \), expressed as \( \mathbf{F} \mathbf{G} \text{en}_i \rightarrow \mathbf{G} \mathbf{F} (\text{en}_i \land c_i) \)
  i.e., “\( i \) enabled almost always executes crit sect infinitely often”
  (Note that \( \mathbf{G} \mathbf{F} (\text{en}_i \land c_i) \) can be replaced by \( \mathbf{G} \mathbf{F} c_i \) 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)
\[ G \neg (P1\_in\_critical\_section \land P2\_in\_critical\_section) \]
specific properties, some related to reachability

- “$\varphi$ never holds in two consecutive states”
  \[ G(\varphi \rightarrow X \neg \varphi) \]

- “if \( \varphi \) holds in state \( s \), then \( \varphi \) holds in all states after \( s \)”
  \[ G(\varphi \rightarrow G \varphi) \]
  why is this different from \( G(\varphi \rightarrow F \varphi) \) ??

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

- “\( \varphi \) holds in at least two states”
  \[ F(\varphi \land X F \varphi) \]

- already seen: “\( \varphi \) holds infinitely often” \( G F \varphi \)

- already seen: “eventually \( \varphi \) always holds” \( F G \varphi \)

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

- “there is no next state”
  \[ \mathbf{X} \bot \]

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

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

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

- “every execution path is finite (system has no infinite execution)”
  \[ \mathbf{F} \mathbf{X} \bot \]

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

- “$\varphi$ holds in every odd state and does not hold in every even state”
  (states are counted from 1)
  $\varphi \land G (\varphi \leftrightarrow X \lnot \varphi)$

- what does the following say:
  $(\varphi \land G (\varphi \leftrightarrow X \lnot \varphi)) \lor X (\varphi \land G (\varphi \leftrightarrow X \lnot \varphi))$ ??

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

  - $\varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \cdots$ (\varphi true in every odd state)
  - $\lnot \varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \cdots$ (\varphi true in every even state)
  - $\varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \lnot \varphi \rightarrow \varphi \rightarrow \cdots$ (\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:

\[
\begin{align*}
\varphi & \rightarrow ??? & \varphi & \rightarrow ??? & \varphi & \rightarrow ??? & \cdots
\end{align*}
\]

- 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

\[
\begin{align*}
\varphi & \rightarrow \varphi & \rightarrow \varphi & \rightarrow \neg \varphi & \rightarrow \varphi & \rightarrow ??? & \cdots
\end{align*}
\]

- 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”
\[ \mathcal{G} \left( \text{request} \rightarrow X F \text{ack} \right) \]

▶ “every request remains true until it is acknowledged”
\[ \mathcal{G} \left( \text{request} \rightarrow (\text{request } U \text{ack}) \right) \]

▶ “every request remains true until it is acknowledged, after which it immediately becomes false”
\[ \mathcal{G} \left( \text{request} \rightarrow ((\text{request } \land \neg \text{ack}) U (\neg \text{request } \land \text{ack})) \right) \]