CS 512, Spring 2018, Handout 09

Model Checking: Examples in LTL

Assaf Kfoury

February 13, 2018

reminder: top-view of model checking (using a temporal logic such as LTL, but not only)

- what we are given :
  1. a transition system TS, which may specify a **protocol** for the simultaneous operation – asynchronous or synchronous – of communicating/interacting processes
  2. a temporal WFF φ expressing some desirable property of TS

- what we want to check :
  1. do **all** paths/traces exhibited by TS satisfy φ?
  2. if we cannot answer preceding question, can we determine whether a “significant” subset of all paths/traces exhibited by TS satisfy φ?
  3. preferably in a fully automated way

- this handout complements Handout 07,
  Practical Patterns of Specification with LTL ,
  which you should review before reading this one.

common properties expressible in LTL

- **safety** "something bad will not happen"
  - $\Box\,\neg(\text{reactor\_temp} > 1000)$
  - $\Box\,\neg((x = 0) \wedge \bigcirc\,(y = z/x))$
  - $\Box\,\neg(\text{system\_crash})$ (the system should never crash)
  - typical form: $\Box\,\neg(\quad\cdots\quad)$

- **liveness** "something good will happen"
  - $\Box\,(\text{start} \to \Diamond\,\text{terminate})$
  - $\Box\,(\text{switch\_on} \to \Diamond\,\text{start})$
  - $\Box\,(\text{switch\_on} \to \bigcirc\,\text{start})$ (perhaps too stringent?)
  - $\Box\,(\text{packet\_sent} \to \Diamond\,\text{packet\_received})$
  - typical form: $\Box\,(\cdots \to \Diamond(\quad\cdots\quad))$ or $\Box\,(\cdots \to \bigcirc\,(\quad\cdots\quad))$

common properties expressible in LTL

- **safety or liveness?** sometimes both
  - "from any state, it is possible to return to a reset state"
    $\square\,(\neg\text{reset} \rightarrow \lozenge\,\text{reset})$
  - "grant a request 3 cycles after receiving the request"
    $\square\,(\text{request} \rightarrow \bigcirc\bigcirc\bigcirc\,\text{grant})$

common properties expressible in LTL

- **fairness**

  "if something is attempted/requested infinitely often,

  then it will be successful/allocated infinitely often"

  - $\Box \Diamond$ ready $\rightarrow \Box \Diamond$ run
  - $\Box \Diamond$ give_one $\rightarrow \Box \Diamond$ receive_one

  - typically $\Box \Diamond (\quad \cdots \quad) \rightarrow \Box \Diamond (\quad \cdots \quad)$
  - fairness w.r.t. a particular φ, the WFF $\Box \Diamond \varphi$ means
    "φ holds infinitely often, if the path is infinite"
    "φ holds at the last state, if the path is finite"

  Remark: We allow paths/traces to be finite in this handout.

- (On the next slide **fairness** is called **strong fairness**)

common properties expressible in LTL

finer examination of fairness [PMC, Definition 5.25, page 258]:
consider many interacting processes, $i = 1, 2, 3, \ldots$, with
en_i = "i is enabled" and c_i = "i is executing critical section"

- **absolute fairness**

  for every $i = 1, 2, \ldots$, expressed as "$\Box \Diamond \mathsf{c}_i$"

  but which ignores that i may not be ready to execute at certain times

- **strong fairness**

  for every $i = 1, 2, \ldots$, expressed as "$\Box \Diamond \mathsf{en}_i \rightarrow \Box \Diamond \mathsf{c}_i$"

  i.e., "i enabled infinitely often, crit sect executed infinitely often"

- **weak fairness**

  for every $i = 1, 2, \ldots$, expressed as "$\Diamond \Box \mathsf{en}_i \rightarrow \Box \Diamond \mathsf{c}_i$"

  i.e., "i enabled almost always, crit sect executed infinitely often"

- more details on *unconditional fairness*, *strong fairness*, and *weak fairness*, in [PMC, Sect. 3.5, pp. 126-140] and handout *Properties of Transition Systems*.

common properties expressible in LTL

- **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**)

  $\Box \neg (\text{P1\_in\_critical\_section} \wedge \text{P2\_in\_critical\_section})$

specific properties, some related to **reachability**

- "φ never holds in two consecutive states"

  $\Box(\varphi \to \bigcirc \neg\varphi)$

- "if φ holds in state s, then φ holds in all states after s"

  $\Box(\varphi \to \Box\varphi)$

  why is this different from $\Box(\varphi \to \Diamond\varphi)$??

- "φ holds in at most one state"

  $\Box(\varphi \to \bigcirc\Box\neg\varphi)$

- "φ holds in at least two states"

  $\Diamond(\varphi \wedge \bigcirc\Diamond\varphi)$

- already seen: "φ holds infinitely often" $\Box\Diamond\varphi$
- already seen: "eventually φ always holds" $\Diamond\Box\varphi$
- "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"

  $(\bigcirc\varphi \to \varphi) \wedge \Box(\bigcirc\bigcirc\varphi \to \varphi \vee \bigcirc\varphi)$

specific properties related to **deadlocks**

- "there is no next state"

  $\bigcirc$ **`false`**

- "every state which has no next state is a **terminal** state"

  $\square (\bigcirc \texttt{false} \rightarrow \text{terminal})$

- "the system is free of deadlocks"

  this is the same as preceding assertion, *i.e.*,

  $\square (\bigcirc \texttt{false} \rightarrow \text{terminal})$

- "a dealock state can be reached" (negation of preceding assertion)

  $\lozenge (\bigcirc \texttt{false} \wedge \neg\text{terminal})$

- "every execution/path is finite (system has no infinite execution)"

  $\lozenge \bigcirc \texttt{false}$

- "every execution/path is infinite (system has no finite execution)"

  $\square \bigcirc \texttt{true}$

specific properties related to **alternation**

- "φ holds in every odd state and does not hold in every even state"

  (assume that states are counted from 1)

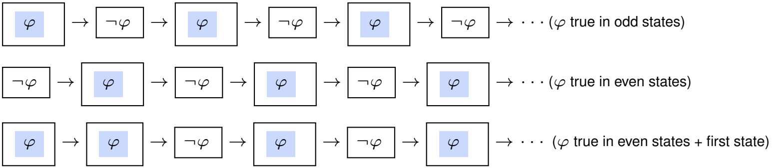
  $\varphi \wedge \Box\,(\varphi \leftrightarrow \bigcirc\,\neg\varphi)$

- what does the following say:

  $(\varphi \wedge \Box\,(\varphi \leftrightarrow \bigcirc\,\neg\varphi)) \;\vee\; \bigcirc\,(\varphi \wedge \Box\,(\varphi \leftrightarrow \bigcirc\,\neg\varphi))$??

- can we replace the preceding WFF by: $\Box\,(\varphi \leftrightarrow \bigcirc\,\neg\varphi)$??

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



specific properties related to **alternation**

- how about the following:
  $(\varphi \wedge \Box(\varphi \leftrightarrow \bigcirc \neg\varphi)) \;\wedge\; \bigcirc(\varphi \wedge \Box(\varphi \leftrightarrow \bigcirc \neg\varphi))$???

  (contradictory WFF, *i.e.*, complicated way of asserting **`false`**)

specific properties related to **alternation**

- suppose we want to express "φ holds in every odd state", *i.e.*,

$$\boxed{\varphi} \to \boxed{??} \to \boxed{\varphi} \to \boxed{??} \to \boxed{\varphi} \to \boxed{??} \to \cdots$$

- can we use $\varphi \wedge \Box\,(\varphi \to \bigcirc\bigcirc\varphi)$??

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

$$\boxed{\varphi} \to \boxed{\varphi} \to \boxed{\varphi} \to \boxed{\neg\varphi} \to \boxed{\varphi} \to \boxed{??} \to \cdots$$

- in fact, "φ holds in every odd state" is NOT expressible in LTL
- describe in English the paths satisfying $\Box\,(\varphi \to \bigcirc\bigcirc\varphi)$
- describe in English the paths satisfying $\varphi \wedge \Box\,(\varphi \to \bigcirc\bigcirc\varphi)$

specific properties related to **responsiveness**

- "every request is eventually acknowledged"

  $\Box\,(\text{request} \rightarrow \bigcirc\,\Diamond\,\text{ack})$

- "every request remains true until it is acknowledged"

  $\Box\,(\text{request} \rightarrow (\text{request}\ \mathbb{U}\ \text{ack}))$

- "every request remains true until it is acknowledged, after which it immediately becomes false"

  $\Box\,(\text{request} \rightarrow ((\text{request} \wedge \neg\text{ack})\ \mathbb{U}\ (\neg\text{request} \wedge \text{ack})))$

(THIS PAGE INTENTIONALLY LEFT BLANK)