CS 512, Spring 2018, Handout 07

Practical Patterns of Specifications with LTL

Assaf Kfoury

7 February 2018

formal semantics of LTL (continuation)

- The satisfaction relation over ω-words $\sigma \in (2^{\mathsf{AP}})^\omega$ is defined in Handout 06:

$$\mathit{Words}(\varphi) \triangleq \left\{ \sigma \in (2^{\mathsf{AP}})^\omega \;\middle|\; \sigma \models \varphi \right\},$$

which is the set of all ω-words in $(2^{\mathsf{AP}})^\omega$ satisfying the LTL formula φ.

formal semantics of LTL (continuation)

- The satisfaction relation over ω-words $\sigma \in (2^{\mathsf{AP}})^\omega$ is defined in Handout 06:

$$\textit{Words}(\varphi) \triangleq \left\{ \sigma \in (2^{\mathsf{AP}})^\omega \;\middle|\; \sigma \models \varphi \right\},$$

  which is the set of all ω-words in $(2^{\mathsf{AP}})^\omega$ satisfying the LTL formula φ.

- [PMC, Def 5.7, page 237]: Let TS $\triangleq (S, \mathsf{Act}, \to, I, \mathsf{AP}, L)$ be a transition system without terminal states, and φ a formula of LTL over AP.
  - The satisfaction relation over (infinite) paths π of TS is defined by:

$$\pi \models \varphi \quad \text{iff} \quad \mathsf{trace}(\pi) \models \varphi$$

  - The satisfaction relation over states s of TS is defined by:

$$\begin{aligned} s \models \varphi \quad & \text{iff} \quad \text{for every path } \pi \text{ starting at } s \text{ we have } \mathsf{trace}(\pi) \models \varphi \\ & \text{iff} \quad \text{for every } \sigma \in \textit{Traces}(s) \text{ we have } \sigma \models \varphi \end{aligned}$$

  - TS satisfies φ, denoted TS $\models \varphi$, iff $\textit{Traces}(\mathsf{TS}) \subseteq \textit{Words}(\varphi)$.

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

- $\boxed{\varphi \triangleq \Box(\text{started} \rightarrow \text{ready})}$

  if $\pi \models \varphi$, then in every state along π, "ready" is true whenever "started" is true

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

- $\boxed{\varphi \triangleq \Box(\text{started} \rightarrow \text{ready})}$

  if $\pi \models \varphi$, then in every state along π, "ready" is true whenever "started" is true

- $\boxed{\varphi \triangleq \Box(\text{requested} \rightarrow \Diamond\ \text{acknowledged})}$

  if $\pi \models \varphi$, then in every state along π, if a "request" (of some resource) occurs, it will eventually be "acknowledged"

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

- $\boxed{\varphi \triangleq \Box(\text{started} \to \text{ready})}$

  if $\pi \models \varphi$, then in every state along π, "ready" is true whenever "started" is true

- $\boxed{\varphi \triangleq \Box(\text{requested} \to \Diamond\ \text{acknowledged})}$

  if $\pi \models \varphi$, then in every state along π, if a "request" (of some resource) occurs, it will eventually be "acknowledged"

- $\boxed{\varphi \triangleq \Box \Diamond \text{enabled}}$

  if $\pi \models \varphi$, then π makes "enabled" true infinitely often

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

- $\boxed{\varphi \triangleq \ \Box(\text{started} \rightarrow \text{ready})}$

  if $\pi \models \varphi$, then in every state along π, "ready" is true whenever "started" is true

- $\boxed{\varphi \triangleq \ \Box(\text{requested} \rightarrow \Diamond \text{ acknowledged})}$

  if $\pi \models \varphi$, then in every state along π, if a "request" (of some resource) occurs, it will eventually be "acknowledged"

- $\boxed{\varphi \triangleq \ \Box \Diamond \text{ enabled}}$

  if $\pi \models \varphi$, then π makes "enabled" true infinitely often

- $\boxed{\varphi \triangleq \ \Diamond \Box \text{ deadlock}}$

  if $\pi \models \varphi$, then π will eventually make "deadlock" continuously true

practical patterns of specifications with LTL [LCS,Sect. 3.2.3]

- $\boxed{\varphi \triangleq \ \Box(\text{started} \rightarrow \text{ready})}$

  if $\pi \models \varphi$, then in every state along π, "ready" is true whenever "started" is true

- $\boxed{\varphi \triangleq \ \Box(\text{requested} \rightarrow \Diamond\ \text{acknowledged})}$

  if $\pi \models \varphi$, then in every state along π, if a "request" (of some resource) occurs, it will eventually be "acknowledged"

- $\boxed{\varphi \triangleq \ \Box \Diamond \text{enabled}}$

  if $\pi \models \varphi$, then π makes "enabled" true infinitely often

- $\boxed{\varphi \triangleq \ \Diamond \Box \text{deadlock}}$

  if $\pi \models \varphi$, then π will eventually make "deadlock" continuously true

- $\boxed{\varphi \triangleq \ \Box \Diamond \text{enabled} \rightarrow \Box \Diamond \text{running}}$

  if $\pi \models \varphi$, then if "enabled" occurs infinitely often along π, then "running" occurs infinitely often along π

practical patterns of specifications with LTL (not in [LCS])

- $\boxed{\varphi \triangleq \; \Box\, \neg(\text{read} \wedge \text{write})}$

  if $\pi \models \varphi$, then in every state along π, not both “read” and “write” are simultaneously true

practical patterns of specifications with LTL (not in [LCS])

- $\boxed{\varphi \triangleq \ \Box\, \neg(\text{read} \wedge \text{write})}$

  if $\pi \models \varphi$, then in every state along π, not both “read” and “write” are simultaneously true

- $\boxed{\varphi \triangleq \ \Box(\text{requested} \rightarrow (\text{requested} \ \mathbb{U} \ \text{granted})}$

  if $\pi \models \varphi$, then in every state along π, if a “request” (of some resource) occurs, then the “request” will persist in every subsequent state until it is “granted”

ω-regular properties *versus* LTL properties

- **Fact:** For every formula φ of LTL (over AP) there exists an NBA $\mathcal{A}_\varphi$ such that
  1. *Words*$(\varphi) = \mathscr{L}(\mathcal{A}_\varphi)$, and
  2. $\mathcal{A}_\varphi$ can be constructed in time and space $2^{\mathcal{O}(n \log n)}$ where $n = |\varphi|$.

ω-regular properties *versus* LTL properties

- **Fact:** For every formula φ of LTL (over AP) there exists an NBA $\mathcal{A}_\varphi$ such that
  1. *Words*$(\varphi) = \mathscr{L}(\mathcal{A}_\varphi)$, and
  2. $\mathcal{A}_\varphi$ can be constructed in time and space $2^{\mathcal{O}(n \log n)}$ where $n = |\varphi|$.

- **Corollary:** Every formula of LTL expresses an ω-regular property.

ω-regular properties *versus* LTL properties

- **Fact:** For every formula φ of LTL (over AP) there exists an NBA $\mathcal{A}_\varphi$ such that
  1. *Words*$(\varphi) = \mathscr{L}(\mathcal{A}_\varphi)$, and
  2. $\mathcal{A}_\varphi$ can be constructed in time and space $2^{\mathcal{O}(n \log n)}$ where $n = |\varphi|$.

- **Corollary:** Every formula of LTL expresses an ω-regular property .

- However, **not** every ω-regular property can be expressed by a formula of LTL .

  Example: There is no formula φ of LTL such that *Words*$(\varphi) = P$ where P is:

  $$P \triangleq \left\{ A_0 A_1 A_2 \cdots \in (2^{\{a\}})^\omega \;\middle|\; a \in A_{2i} \text{ for every } i \geqslant 0 \right\}.$$

  But there exists an NBA $\mathcal{A}$ such that $\mathscr{L}(\mathcal{A}) = P$.
  (Why? See Problem 4 in Assignment #2.)

many properties **not** expressible in LTL [LCS,Sect 3.2.3]

Many properties of interest assert ***the existence of a path*** satisfying a certain condition, and such properties cannot be expressed in LTL. Examples of such properties:

- *from every state it is possible to reach a **reset** state, i.e., for every state s, **there is** a path from s that enters a state s' where "reset" is true.*
- *one possible behavior of the elevator is to remain idle on the third floor, i.e., from the state in which it is on the third floor, **there is** a path that keeps it there.*

(THIS PAGE INTENTIONALLY LEFT BLANK)