A Taxonomy of LT properties

Invariant $\subset$ Safety "nothing bad ever happens" Liveness "something good eventually happens"

An LT property P_{inv} is an invariant if there is a propositional formula φ s.t. for all $\sigma \in P_{inv}$ and all $i \geq 0$, $\sigma[i] \models \varphi$

Given that

$$
\begin{aligned}
TS \models P_{inv} &\iff Traces(TS) \subseteq P_{inv} \\
&\iff trace(\pi) \in P_{inv} \quad \forall \pi \text{ path of } G(TS) \\
&\iff L(s) \models \phi \quad \forall s \text{ on a path of } G(TS) \\
&\iff \boxed{L(s) \models \phi \quad \forall s \in Reach(TS)}
\end{aligned}
$$

all reachable state of TS satisfy ϕ

we can conclude that an invariant is a "state-property"; in fact, invariant properties can be linearly checked on transition systems whose state graph is finite.

Exercise Show that P_{mutex} is an invariant $\quad \phi = \neg crit_1 \vee \neg crit_2$

Safety

In general safety properties impose conditions on finite path fragments of executions e.g.

"before withdrawing money, a correct PIN is entered" (*)

Intuition: an infinite execution violating (*) has a finite prefix violating it

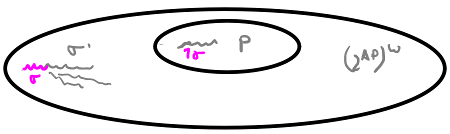
for $\sigma = \sigma_0 \dots \sigma_n \sigma_{n+1} \dots \qquad \sigma_{<n} = \sigma_0 \dots \sigma_n \; ; \quad \sigma_{<0} = \varepsilon$

$pref(\sigma) = \bigcup_{n \in \omega} \sigma_{<n}$

"bad Prefix"

Safety P_{safe} $\forall \sigma \in (2^{AP})^\omega \setminus P_{safe} \; \exists n \geq 0 : \sigma_{<n} (2^{AP})^\omega \cap P_{safe} = \emptyset$

$BadPref(P) = \{ \sigma \in (2^{AP})^* \mid \exists \sigma' \in (2^{AP})^\omega \setminus P : \sigma \in pref(\sigma') \wedge \sigma (2^{AP})^\omega \cap P = \emptyset \}$



Lemma $TS \models P_{safe} \iff Traces_{fin}(TS) \cap BadPref(P_{safe}) = \emptyset$

$:= \bigcup_{s \in S} Traces_{fin}(s) := trace(Paths_{fin}(s))$

set of finite path fragments on $G(TS)$

Proof $(\Rightarrow)$ If $\hat{\sigma} \in Traces_{fin}(TS) \cap BadPref(P_{safe})$

$\Rightarrow \exists\, \sigma \in Traces(TS), n \geq 0 : \hat{\sigma} = \sigma_{<n}$

$\Rightarrow \sigma \notin P_{safe}$

$\Rightarrow TS \not\models P_{safe}$

$(\Leftarrow)$ If $TS \not\models P_{safe} \overset{def}{\iff} \exists\, \sigma \in Traces(TS) : \sigma \notin P_{safe}$

$\Rightarrow \exists\, n \geq 0 : \sigma_{<n} \in BadPref(P_{safe})$

$\Rightarrow \sigma_{<n} \in Traces_{fin}(TS) \cap BadPref(P_{safe})$ $\square$

weaker than full trace inclusion
$\Rightarrow$ good to show that refinement is ok

Thm $Traces_{fin}(TS) \subseteq Traces_{fin}(TS') \iff$

$\forall$ safety properties P $\quad TS' \models P \Rightarrow TS \models P$

Proof $(\Rightarrow)$ P is a safety prop $\overset{L}{\Longrightarrow} Traces_{fin}(TS') \cap BadPref(P) = \emptyset$

$\overset{hyp}{\Longrightarrow} Traces_{fin}(TS) \cap BadPref(P) = \emptyset \overset{L}{\iff} \checkmark$

$(\Leftarrow)$ Take $P = closure(Traces(TS'))$

P is a safety property and $TS' \models P$

$\overset{hyp}{\Longrightarrow} Traces(TS) \subseteq P \Rightarrow pref(Traces(TS)) \subseteq pref(P)$

$\wedge\; Traces_{fin}(TS) = pref(Traces(TS))$

$\subseteq pref(P) = pref(Traces(TS')) = Traces_{fin}(TS')$ $\square$

Given an LT prop. P, $\quad closure(P) = \{ \sigma \in (2^{AP})^{\omega} \mid pref(\sigma) \subseteq pref(P) \}$

Exercise: show that a prop P is safe $\iff P = closure(P)$

Safety "constraints" finite behaviour while liveness imposes conditions on infinite behaviour

Liveness P_{live} $\quad \forall w \in (2^{AP})^* \; \exists \sigma \in (2^{AP})^\omega : \; w\sigma \in P_{live}$

$$\Updownarrow$$

$$pref(P_{live}) = (2^{AP})^*$$

"something good happens"

Exercise Give the properties informally specified as

1. "each process eventually enters the critical section"
2. "each process enters the critical section infinitely often"
3. "each waiting process eventually enters the critical section"

$$P_3 = \{\sigma \in (2^{AP})^\omega \mid \bigwedge_{1 \le h \le n} \forall i \ge 0 : w_h \in \sigma[i] \Rightarrow \exists j > i : c_h \in \sigma[j]\}$$

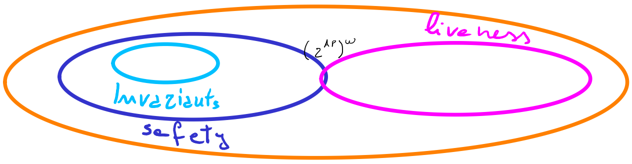
$$P_2 = \{\sigma \in (2^{AP})^\omega \mid \bigwedge_{1 \le h \le n} \forall i \ge 0 \; \exists j > i : c_h \in \sigma[j]\}$$

$$P_1 = \{\sigma \in (2^{AP})^\omega \mid \bigwedge_{1 \le h \le n} \exists j \ge 0 : c_h \in \sigma[j]\}$$

there are LT prop that are neither safety nor liveness prop., but:

Decomposition theorem $\forall$ LT prop P $\exists P_s$ safety, P_ℓ liveness : $P = P_s \cap P_\ell$

LT prop.



126 *Linear-Time Properties*

from [1]

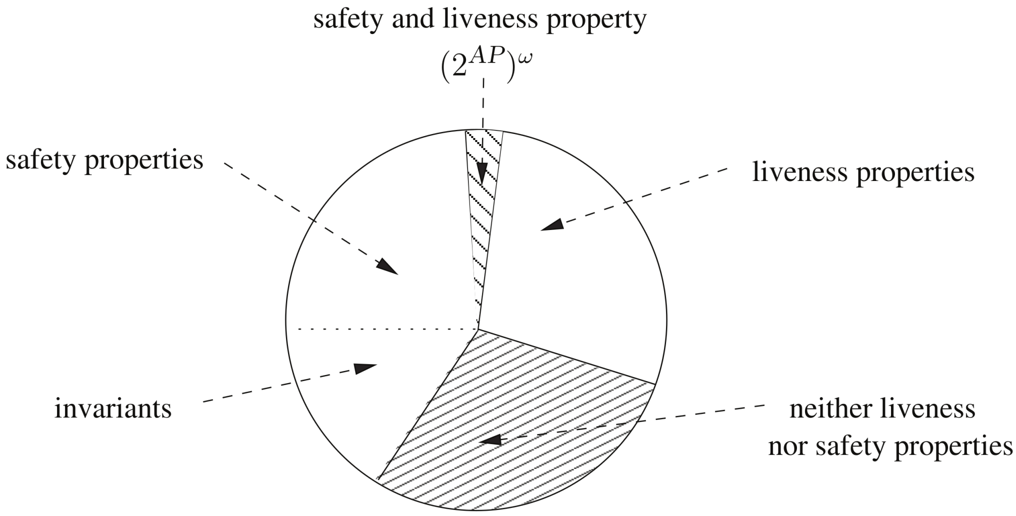


Figure 3.11: Classification of linear-time properties.

Fairness

Usually liveness properties do not hold, unless fairness assumptions are made

Example 3.44. A Simple Shared-Variable Concurrent Program

Consider the following two processes that run in parallel and share an integer variable x that initially has value 0:

$$\textbf{proc}\ \mathsf{Inc} \;=\; \textbf{while}\ \langle x \geqslant 0\ \textbf{do}\ x := x+1 \rangle\ \textbf{od}$$
$$\textbf{proc}\ \mathsf{Reset} \;=\; x := -1$$

The pair of brackets $\langle \ldots \rangle$ embraces an atomic section, i.e., process Inc performs the check whether x is positive and the increment of x (if the guard holds) as one atomic step. Does this parallel program terminate? When no fairness constraints are imposed, it is possible that process Inc is permanently executing, i.e., process Reset never gets its turn, and the assignment $x = -1$ is not executed. In this case, termination is thus not guaranteed, and the property is refuted. If, however, we require unconditional process fairness, then every process gets its turn, and termination is guaranteed. ■

there is a wealth of fairness conditions

$A \subseteq Act$

An execution (fragment) $\rho = s_0 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \cdots$ is

- unconditionally A-fair $\quad \overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$

  "Every process gets its turn infinitely often"

$:= \{\alpha \in Act \mid \exists s' \in S : s_j \xrightarrow{\alpha} s'\}$

- strongly A-fair $\quad \overset{\infty}{\exists} j \geq 0 : A \cap Act(s_j) \neq \emptyset \Rightarrow \overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$

  "Every process enabled infinitely often gets its turn infinitely often"

- weakly A-fair $\quad \overset{\infty}{\forall} j \geq 0 : A \cap Act(s_j) \neq \emptyset \Rightarrow \overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$

  "Every process continuously enabled from a certain time instant on gets its turn infinitely often"

Checking liveness property is often made by restricting to fair executions: $\boxed{TS \models_F P \iff FairTraces(TS) \subseteq P}$

Here, $\overset{\infty}{\exists} j$ stands for "there are infinitely many j" and $\overset{\infty}{\forall} j$ for "for nearly all j" in the sense of "for all, except for finitely many j". The variable j, of course, ranges over the natural numbers.