$TS = \langle S, Act, \to, I, AP, L \rangle$ transition system

$G(TS) = \langle S, E \rangle$ state graph of TS

$E ::= \{ \langle s, s' \rangle \in S \times S \,/\, \underbrace{s \xrightarrow{a} s' \text{ for some } a \in Act}_{s' \in Post(s)} \}$

path-fragments

paths

traces

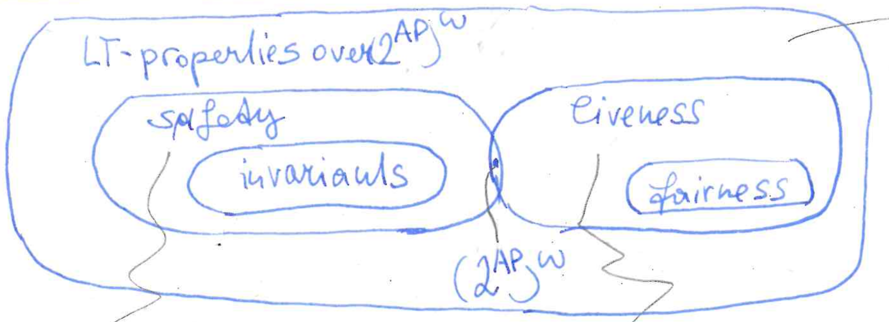
Linear-time properties (LT-property) : $P \subseteq (2^{AP})^{\omega}$ i.e. $P \in 2^{(2^{AP})^{\omega}}$

transition system satisfies P

$TS \models P \iff Traces(TS) \subseteq P$.

state s satisfies P

$s \models P \iff Traces(s) \models P$

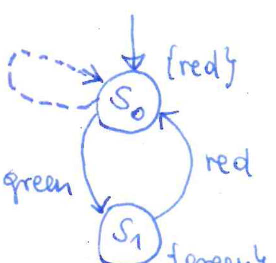


neither fairness nor liveness properties, but intersections of fairness and liveness properties.

"nothing bad ever happens"

"something good is eventually / keeps happening"

$AP = \{red, green\}$



$\boxed{TS}$

execution fragment : s_0 green s_1 red s_0

execution : s_0 green s_1 red s_0 green ...

path fragment : $s_0 s_1 s_0$, $s_1 s_0$

path : $s_0 s_1 s_0 s_1 \dots$

trace : $\{red\}\{green\}\{red\}\{green\} \dots$

LINEAR-TIME BEHAVIOUR & PROPERTIES

$TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ transition system

$G(TS) = \langle V, E \rangle$ with $V := S$ and $E := \{\langle s, s' \rangle \in S \times S \mid \underbrace{s \xrightarrow{a} s' \text{ for some } a \in Act}_{s' \in Post(s)}\}$

STATE GRAPH of TS

A PATH FRAGMENT is a (finite or infinite) state sequence ($\pi = s_0 s_1 s_2 \ldots$) on its state graph:

$\pi \in S^* \cup S^\omega$ such that $\forall 0 \le i < |\pi| : \pi[i+1] \in Post(\pi[i])$.

NOTATION: let $\pi = s_0 s_1 s_2 \ldots$. $\pi[j] := s_j$ $\quad \pi[..j] := s_0 s_1 \ldots s_j$

$first(\pi) := s_0$, $\quad \pi[j..] := s_j s_{j+1} \ldots$

$\boxed{Paths_{fin}(TS)}$ if π is finite, $\pi = s_0 s_1 \ldots s_n$, then: $Last(\pi) := s_n$, $len(\pi) := n$

$\boxed{Paths(TS)}$ if π is infinite, $\pi = s_0 s_1, \ldots$, then: $Last(\pi)\uparrow$, $len(\pi) := \omega$.

π is **maximal** if $\underbrace{\pi \in S^* \,\&\, Post(Last(\pi)) = \emptyset}_{\pi \text{ is finite and ends in a terminal state}}$ or $\underbrace{\pi \in S^\omega}_{\pi \text{ is infinite}}$.

π is **initial** if $\pi[0] \in I$

π is a **path** if π is initial & maximal.

TRACE of π (path fragment) is $\{L(\pi[i])\}_{0 \le i < |\pi|} =: trace(\pi) \subseteq (2^{AP})^{|\pi|}$

$traces(\Pi) := \{trace(\pi) \mid \pi \in \Pi\}$ for every set Π of path fragments

$Traces(s) := traces(Paths(s))$ for every $s \in S$

where $Paths(s)$ are all maximal path fragments from s in TS

$Traces(TS) := \bigcup_{s \in I} Traces(s)$

A LINEAR-TIME PROPERTY P over set AP of atomic propositions is a subset of $(2^{AP})^\omega$ [i.e. an infinite sequence of subsets of prop's]

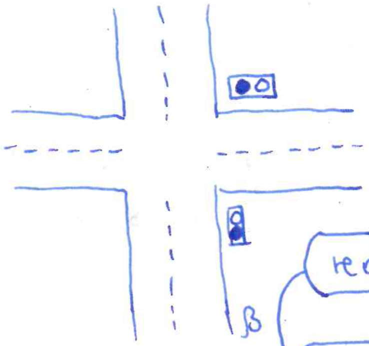
i.e. $P \subseteq (2^{AP})^\omega$

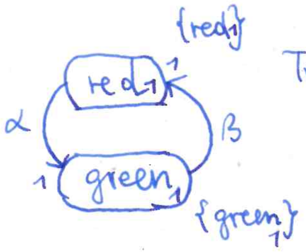
Transition system TS satisfies $P \subseteq (2^{AP})^\omega$ if $Traces(TS) \subseteq P$.

NOTATION $TS \models P$

State $s \in S$ satisfies $P \subseteq (2^{AP})^\omega$ if $Traces(s) \subseteq P$.

NOTATION $s \models P$

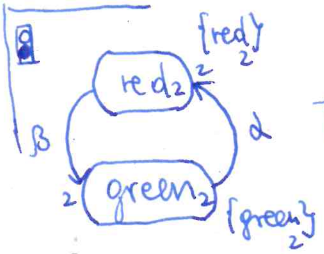




TrLight$_1$

$$\frac{x \xrightarrow{a}_1 x' \quad y \xrightarrow{a}_2 y'}{x \| y \xrightarrow{a} x' \| y'}$$

$\boxed{\text{handshaking communication}}$



TrLight$_2$

We consider

TrLight$_1$ || TrLight$_2$

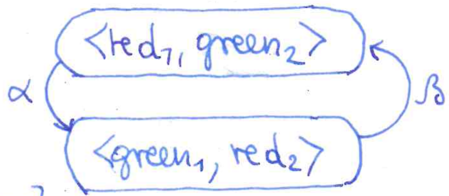
$red_1 \| green_2 \xrightarrow{\alpha} green_1 \| red_2$

$\xrightarrow{\beta} red_1 \| green_2$

$$\text{TrLight}_i = \langle \underbrace{\{red_i, green_i\}}_{S_i}, \underbrace{\{\alpha, \beta\}}_{Act}, \rightarrow_i, \underbrace{\{red_i, green_i\}}_{I_i}, \underbrace{\{red_i, green_i\}}_{AP_i}, L_i \rangle$$

$L_i(red_i) := \{red_i\}$

$L_i(green_i) := \{green_i\}$



$$\text{TrLights} = \text{TrLight}_1 \| \text{TrLight}_2 = \langle S_1 \times S_2, Act, \rightarrow, I_1 \cup I_2, AP_1 \cup AP_2, L \rangle$$

$L(\langle s_1, s_2 \rangle) = L(s_1) \cup L(s_2)$

Path fragments:

in TrLight$_1$: $red_1\, green_1\, red_1$ (initial, not maximal / path), $green_1\, red_1\, green_1\, red_1 \ldots$ (maximal / initial / path)

in TrLight$_1$ || TrLight$_2$:

$\langle red_1, red_2 \rangle$ initial / maximal / path

$\langle green_1, red_2 \rangle \langle red_1, green_2 \rangle$ initial / not maximal / path

$\langle red_1, green_2 \rangle \langle green_1, red_2 \rangle \langle red_1, green_2 \rangle \ldots$ initial / maximal / path.

Traces:

in TrLight$_1$: $\{red_1\}\{green_1\}, \{red_1\}$ trace of ~~of~~ a not maximal path fragment

LT-properties: P: the first traffic light is green infinitely often

$\{red_1, green_2\}\{green_1, red_2\}\{red_1, green_2\} \ldots \in P$

$\emptyset \{green_1\} \emptyset \{green_1\} \emptyset \{green_1\} \ldots \in P$

$\{green_1, green_2\}\{green_1, green_2\} \ldots \in P$

The importance of traces

$\langle S, Act, \rightarrow, I, AP, L \rangle$ ($\rightarrow \subseteq S \times Act \times S$, $I \subseteq S$, $L: S \rightarrow 2^{AP}$)

WLOG: no terminal states in TS (hence all maximal path fragments are infinite)

The trace of a maximal path fragment of TS is $trace(\pi) := \{\underbrace{L(\pi[i])}_{\in 2^{AP}}\}_{i \geq 0}$.

$$TS \models P \;:\Leftrightarrow\; Traces(TS) \subseteq P, \quad \text{for all } P \subseteq (2^{AP})^\omega$$
$$s \models P \;:\Leftrightarrow\; Traces(s) \subseteq P, \quad \text{―――――――}$$

TS_1, TS_2 two transition systems over the same set AP of atomic propositions

Then: $\underset{\text{refinement}}{Traces(TS_1)} \subseteq \underset{\text{abstract model}}{Traces(TS_2)}$ could mean: "TS_1 implements TS_2 (correctly)".

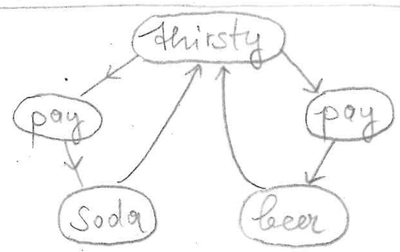
Theorem. $Traces(TS_1) \subseteq Traces(TS_2) \Leftrightarrow \forall$ LT-properties P over AP $[TS_2 \models P \Rightarrow TS_1 \models P]$

Proof. ($\Rightarrow$) Suppose $Traces(TS_1) \subseteq Traces(TS_2)$ (assm).
Let $P \subseteq (2^{AP})^\omega$ be an LT-property over AP.
Then: $TS_2 \models P \underset{\text{by def.}}{\Longrightarrow} Traces(TS_2) \subseteq P$
$\underset{\text{by assm}}{\Longrightarrow} Traces(TS_1) \subseteq Traces(TS_2) \subseteq P$
$\underset{\text{by def.}}{\Longrightarrow} TS_1 \models P$

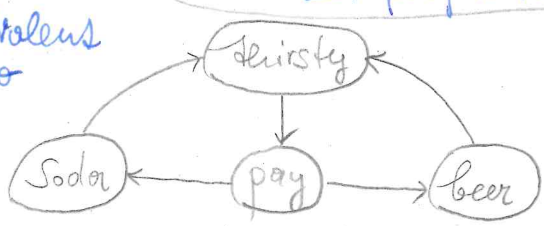
($\Leftarrow$) Suppose $TS_2 \models P \Rightarrow TS_1 \models P$ holds for all LT-properties over AP.
Then it also holds for $P := Traces(TS_2)$. As $TS_2 \models Traces(TS_2)$ holds obviously (as it means $Traces(TS_1) \subseteq Traces(TS_2)$), we conclude $TS_1 \models Traces(TS_2)$, which means: $Traces(TS_1) \subseteq Traces(TS_2)$.

Corollary. $\underbrace{Traces(TS_1) = Traces(TS_2)}_{\text{trace-equivalence of } TS_1, TS_2} \Leftrightarrow \forall$ LT-properties over AP $\underbrace{[TS_1 \models P \Leftrightarrow TS_2 \models P]}_{TS_1, TS_2 \text{ fulfill the same LT properties}}$.

Example.



equivalent to



Taxonomy of LT-properties

Invariant $\subset$ Safety "nothing bad ever happens" Liveness "something good is eventually / keeps happening"

An LT-property P_{inv} over AP (i.e. $P_{inv} \subseteq 2^{AP}$) is an invariant:

$$:\Leftrightarrow \quad P_{inv} = \{A_0 A_1 A_2 \ldots \in 2^{AP} \mid A_i \models \Phi\}$$ for some formula Φ of propositional calculus over AP.

Example: $AP = \{a, b, c\}$, $\Phi = a \vee b$

$P_1 := \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid A_i \models \Phi \text{ for all } i \geq 0\}$

$= \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid (a \in A_i \text{ or } b \in A_i) \text{ for all } i \geq 0\}$

For such a property P_{inv} given by a prop. formula Φ:

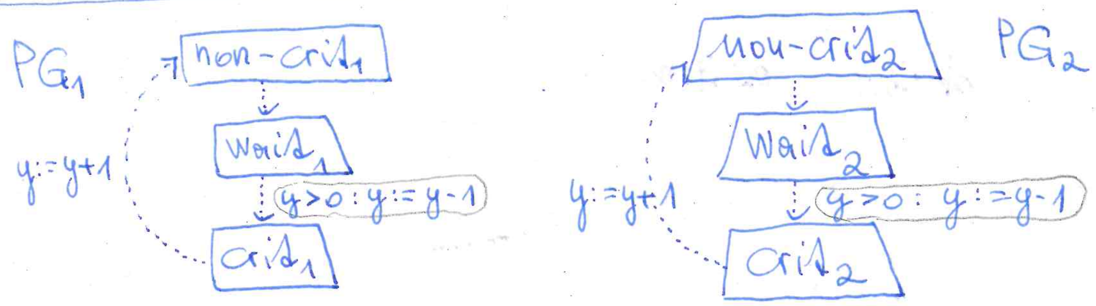
$TS \models P_{inv} \Leftrightarrow Traces(TS) \subseteq P_{inv}$

$\Leftrightarrow \forall \pi$ path of $G(TS)$: $trace(\pi) \in P_{inv}$

$\Leftrightarrow \forall s$ state of TS on a path π of TS: $s \models \Phi$

$\Leftrightarrow \forall s$ reachable state of TS: $s \models \Phi$.

Thus invariants are state-properties that can be decided for a transition system by checking them in every state.



Over $AP = \{crit_i, wait_i, non\text{-}crit_i \mid i \in \{1,2\}\}$ the desired property Φ of a mutual exclusion algorithm can be expressed by the invariant: $\neg crit_1 \vee \neg crit_2$.

Safety

Safety properties impose conditions on <u>finite path fragments</u> of executions

e.g. "before withdrawing money, a correct PIN is entered" ⊛
at an ATM $\quad BP = (\neg p^+ w)^* w(true)^*$

<u>Intuition:</u> an infinite execution violating ⊛ has a finite prefix that already violates ⊛

P_{safe} is a <u>safety property</u> $:\Leftrightarrow \exists \underbrace{BP}_{\text{bad prefixes}} \subseteq (2^{AP})^*.\ P_{safe} = (2^{AP})^\omega \setminus \underbrace{(BP.(2^{AP})^\omega)}_{\text{bad traces have bad prefixes}}$

$\underset{(book)}{\Leftrightarrow} \forall \sigma \in (2^{AP})^\omega \setminus P_{safe}.\ \exists n \geq 0.\ \left[(\sigma_{<n}.(2^{AP})^\omega) \cap P_{safe} = \phi \right]$.

(every "unsafe" trace has a bad prefix)

$BP(P_{safe}) = \{\sigma_0 \in (2^{AP})^* \,/\, \sigma_0.(2^{AP})^\omega \cap P_{safe} = \phi\}$

<u>Lemma.</u> $TS \models P_{safe} \Leftrightarrow Traces_{fin}(TS) \cap BP(P_{safe}) = \phi$

Proof. $(\Rightarrow)$ If $\sigma_0 \in Traces_{fin}(TS) \cap BP(P_{safe})$

we proceed indirectly

$\Rightarrow \exists \sigma \in Traces(TS)\ \exists n\ (\sigma_{<n} = \sigma_0 \wedge \underbrace{\sigma_0.(2^{AP})^\omega}_{\sigma\in} \cap P_{safe} = \phi)$

$\Rightarrow \exists \sigma \in Traces(TS).\ \sigma \notin P_{safe}$

$\Rightarrow Traces(TS) \not\subseteq P_{safe}$

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

$(\Leftarrow)$ If $TS \not\models P_{safe}$ $\Rightarrow Traces(TS) \not\subseteq P_{safe}$

we proceed indirectly

$\Rightarrow \exists \sigma \in Traces(TS)\ \ \sigma \notin P_{safe}$

$\Rightarrow \exists \sigma \in Traces(TS)\ \exists n \geq 0.\ \sigma_{<n} \in BP(P_{safe})$

$\Rightarrow \exists \sigma \in Traces(TS)\ \exists n\ \ \sigma_{<n} \in Traces_{fin}(TS) \cap BP(P_{safe})$

$\Rightarrow Traces_{fin}(TS) \cap BP(P_{safe}) \neq \phi$.

<u>Thm</u> $Traces_{fin}(TS_1) \subseteq Traces_{fin}(TS_2) \Leftrightarrow \forall$ safety prop. $P\ (TS_2 \models P \Rightarrow TS_1 \models P)$

Proof. $(\Rightarrow)$ Let P be a safety property, and (hyp) $Traces_{fin}(TS_1) \subseteq Traces_{fin}(TS_2)$

Then: $TS_2 \models P \overset{(L)}{\Leftrightarrow} Traces_{fin}(TS_2) \cap BP(P_{safe}) = \phi$

$\Rightarrow Traces_{fin}(TS_1) \cap BP(P_{safe}) = \phi$

$\overset{(L)}{\Leftrightarrow} TS_1 \models P$.

Thm (%) $\text{Traces}_{fin}(TS_1) \subseteq \text{Traces}_{fin}(TS_2) \Leftrightarrow$

$\Leftrightarrow \forall P$ safety property: $TS_2 \models P \Rightarrow TS_1 \models P$

$(\Leftarrow)$ Lemma. (i) P is safety property $\Leftrightarrow P = \text{closure}(P)$ $=: \{\sigma \in (2^{AP})^\omega \mid \text{pref}(\sigma) \subseteq \text{pref}(P)\}$

(ii) $\text{closure}(\text{closure}(P)) = \text{closure}(P)$ for all $P \subseteq (2^{AP})^\omega$

For showing "$\Leftarrow$", we assume that

(hyp) $(TS_2 \models P \Rightarrow TS_1 \models P)$ holds for all safety properties P.

We let $P := \text{closure}(\text{Traces}_{fin}(TS_2))$. Then P is a safety property by the Lemma. Also $TS_2 \models P$ because:

$\text{Traces}(TS_2) \subseteq \text{closure}(\text{Traces}_{fin}(TS_2)) = P$.

Then $TS_1 \models P$ by (hyp), and therefore $\text{Traces}(TS_1) \subseteq P$.

Now we conclude:

$$\begin{aligned} \text{Traces}_{fin}(TS_1) &= \text{pref}(\text{Traces}(TS_1)) \\ &\subseteq \text{pref}(P) \\ &= \text{pref}(\text{closure}(\text{Traces}_{fin}(TS_2))) \\ &= \text{Traces}_{fin}(TS_2). \end{aligned}$$

Corollary. $\text{Traces}_{fin}(TS_1) = \text{Traces}_{fin}(TS_2) \Leftrightarrow$

$\Leftrightarrow \forall P$ safety property: $TS_2 \models P \Rightarrow TS_1 \models P$.

Prop. Every invariant is a safety property.

$P = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid A_i \models \Phi \text{ for all } i \geq 0\}$ for some propositional formula Φ over AP

$= (2^{AP})^\omega \setminus (mBP . (2^{AP})^\omega)$

where $mBP = \{A_0 A_1 A_2 \ldots A_{n-1} A_n \mid A_i \models \Phi \text{ for all } i \in \{0,1,\ldots,n-1\}, A_n \not\models \Phi, \text{ where } n \geq 0\}$

minimal bad prefixes

Example. vending machine gives 3 soda initially $AP = \{\text{beer}, \text{soda}\}$

$P_{3\text{-soda}} = \{\{\text{soda}\}\{\text{soda}\}\{\text{soda}\} A_3 A_4 \ldots \mid A_3 A_4 \ldots \subseteq AP\}$

is a safety property

$BP = (2^{AP} \setminus \{\text{soda}\})(2^{AP})^* + \{\text{soda}\} . (2^{AP} \setminus \{\text{soda}\})(2^{AP})^*$

$+ \{\text{soda}\}\{\text{soda}\}(2^{AP} \setminus \{\text{soda}\})(2^{AP})^*$

$mBP = (2^{AP} \setminus \{\text{soda}\}) + \{\text{soda}\} . (2^{AP} \setminus \{\text{soda}\}) + \{\text{soda}\}\{\text{soda}\}(2^{AP} \setminus \{\text{soda}\})$

$AP = \{red, green, yellow\}$ traffic light

P_1: "at least one light is always on"

$$P_1 = \{ A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid \underbrace{|A_i| \geq 1}_{\Leftrightarrow A_i \neq \phi} \text{ for all } i \in \mathbb{N} \}$$

$$BP = \{ A_0 \ldots A_n \in (2^{AP})^* \mid n \geq 0,\ A_i = \phi \text{ for some } i \in \{0, \ldots, n\} \}$$

P_2: "it is never the case that 2 lights are switched on at the same time"

$$P_2 = \{ A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid |A_i| \leq 1 \text{ for all } i \in \mathbb{N} \}$$

$$BP = \{ A_0 \ldots A_n \in (2^{AP})^* \mid n \geq 0,\ |A_i| \geq 2 \text{ for } i \in \{0 \ldots n\} \}$$

P_3: "a red phase must immediately be preceded by a yellow phase"

$$BP(P_3) = \{ A_0 A_1 \ldots A_n \in (2^{AP})^* \mid \exists\, 0 \leq i < n.\ A_i \not\ni yellow \text{ and } A_{i+1} \ni red \}$$

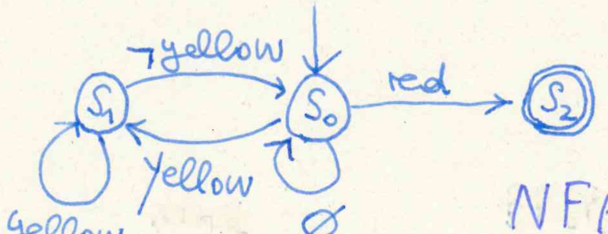
$$P_3 = \{ A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid \forall i \geq 0.\ (red \in A_i \Rightarrow i > 0 \wedge A_{i-1} \ni yellow) \}$$

$\phi \phi \{red\}$, $\phi \{red\} \in BP(P_3)$

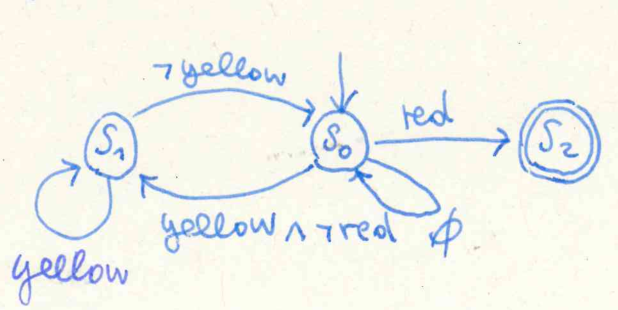
$\{yellow\} \{yellow\} \{red\} \{red\} \phi \{red\} \in BP(P_3)$

$\notin mBP(P_3)$

not a minimal bad prefix



NFA for $BP(P_3)$



DFA for $mBP(P_3)$

Beverage vending machine: "the house never Looses"

P: "The number of inserted coins is always at least the number of dispensed drinks" $AP := \{pay, drink\}$

$$P = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \ / \ \text{forall } i \geq 0: |\{0 \leq j \leq i \ / \ pay \in A_j\}| \geq |\{0 \leq j \leq i \ / \ drink \in A_j\}| \}$$

bad prefixes: $\left\{ \begin{array}{l} \emptyset \{pay\} \{drink\} \{drink\} \notin P \\ \emptyset \{pay\} \{drink\} \emptyset \{pay\} \{drink\} \{drink\} \notin P \end{array} \right.$

Safety constrains finite behaviour while liveness constrains infinite behaviour

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

"something good happens (always) eventually"

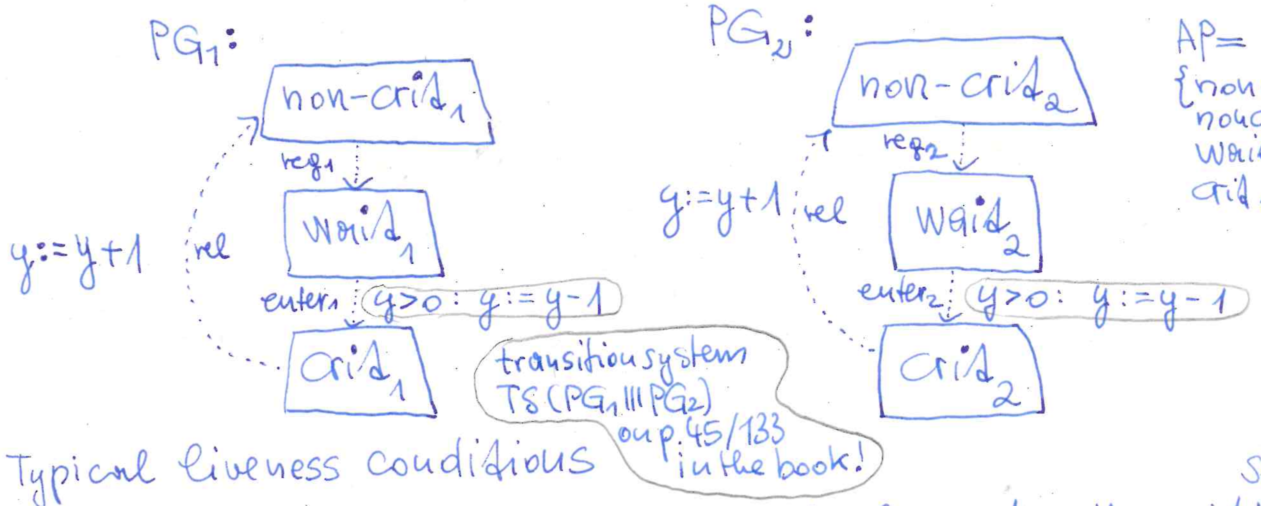
$$\Updownarrow$$

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

$$\Updownarrow$$

$$\underbrace{(2^{AP})^* . P_{live} = P_{live}}$$

Exercise. Semaphore-based mutual exclusion.



$AP = \{noncrit_1, noncrit_2, wait_1, wait_2, crit_1, crit_2\}$

Typical liveness conditions

- (eventually) each process will eventually enter its critical section

  $P_1 = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \ / \ (\exists j \geq 0.\ crit_1 \in A_j) \wedge (\exists j \geq 0.\ crit_2 \in A_j)\}$

- (repeated eventually) each process will enter its critical section infinitely often

  $P_2 = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \ / \ \forall k \geq 0\ \exists j \geq k\ (crit_1 \in A_j) \wedge \underbrace{\forall k \geq 0\ \exists j \geq k\ (crit_2 \in A_j)}_{\overset{\infty}{\exists} j \geq 0.\ crit_2 \in A_j}\}$

- (starvation freedom) each waiting process will eventually enter its critical section

  $P_3 = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \ / \ \forall k \geq 0\ (wait_1 \in A_k \Rightarrow \exists j > k.\ crit_1 \in A_j) \wedge \forall k \geq 0\ (wait_2 \in A_k \Rightarrow \exists j > k.\ crit_2 \in A_j)\}$

if we would not have PG_1, PG_2 but processes that could stop waiting without going in their critical sections, then a stronger formulation could be:

$P_3' = \{A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \ / \ \neg \exists k \geq 0\ \forall j \geq k\ (wait_1 \in A_{j+k} \wedge crit_1 \notin A_{j+k}) \wedge \neg \exists k \geq 0\ \forall j \geq k\ (wait_2 \in A_{j+k} \wedge crit_2 \notin A_{j+k})\}$

Safety versus Liveness Properties

- Are safety and liveness properties disjoint? (No)
- Is any LT-property a safety or liveness property? (No)

<u>Lemma.</u> The single LT-property over AP that is <u>both</u> a safety and a liveness property is $(2^{AP})^\omega$.

Proof. Let P be a liveness property over AP. Then $\mathrm{pref}(P) = (2^{AP})^*$. It follows that $\mathrm{closure}(P) = (2^{AP})^\omega$. If P is a safety property, too, then $P = (2^{AP})^\omega$.

<u>Example.</u> $\left.\begin{array}{l}\text{"vending machine provides beer infinitely often}\\ \text{after initially providing soda three times in a row."}\end{array}\right\} P$

$$P = P_{3\text{-soda}} \cap P_{\infty\text{-beer}}$$

AP = {soda, beer}

safety property

$P_{3\text{-soda}} := \{ \{soda\}\{soda\}\{soda\} A_3 A_4 \ldots \mid A_j \subseteq AP \text{ for } j \geq 3 \}$

$P_{\infty\text{-beer}} := \{ A_0 A_1 A_2 \ldots \in (2^{AP})^\omega \mid \overset{\infty}{\exists} j \geq 0.\ A_j = \{beer\} \}$

liveness property

$BP = (2^{AP} \setminus \{soda\})(2^{AP})^*$
$+ \{soda\}(2^{AP} \setminus \{soda\})^*$
$+ \{soda\}\{soda\}(2^{AP} \setminus \{soda\})^*$

<u>Theorem.</u> (Decomposition) For every LT-property P over AP there exists a safety property P_{safe} and a liveness property P_{live} such that $P = P_{safe} \cap P_{live}$.

Namely: $P_{safe} := \mathrm{closure}(P)$,
$P_{live} := P \cup ((2^{AP})^\omega \setminus \mathrm{closure}(P))$

Topological characterization:

metric d on $(2^{AP})^\omega$: $d(\sigma_1, \sigma_2) = \begin{cases} 0 & \ldots\ \sigma_1 = \sigma_2 \\ \frac{1}{2^n} & \ldots\ \sigma_1 \neq \sigma_2 \text{ and } n \text{ is the shortest common prefix of } \sigma_1 \text{ and } \sigma_2 \end{cases}$

induces topology $\mathcal{T}_d$

in $\langle (2^{AP})^\omega, \mathcal{T}_d \rangle$: closed sets ~ safety properties
dense sets ~ liveness properties
closure(P) ~ topological closure of P

The decomposition theorem then follows from:

<u>Proposition:</u> $\langle X, \mathcal{T} \rangle$ topological space. For all sets $A \subseteq X$ there exists a dense set $D \subseteq X$ such that $A = \overline{A} \cap D$.

Proof. Let $D := A \cup (X \setminus \overline{A})$.

Fairness

Usually liveness properties cannot be guaranteed without some assumptions about fairness.

Process fairness: A server S for processes $P_1, \dots, P_N$ should answer any continuous request eventually.

Starvation freedom: e.g. mutual exclusion algorithms
"Once access is requested by a process, it is not kept waiting forever."
"Each process is infinitely often in its critical section."

$A \subseteq Act$ path $\pi = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_3 \longrightarrow \dots$

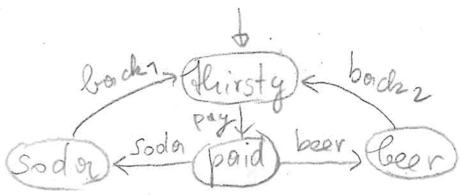
An execution fragment $\rho = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} s_3 \xrightarrow{\alpha_4} \dots$ is

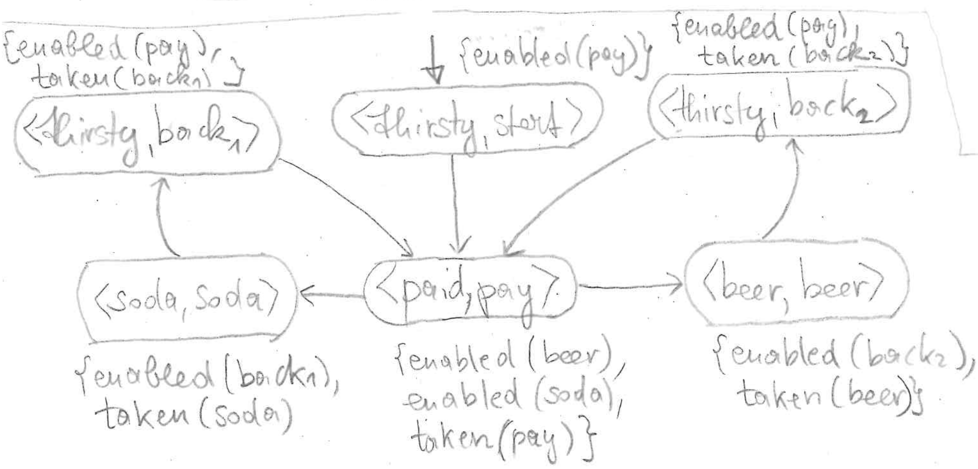
taken$(s_j) \in$ taken(A)

- unconditionally A-fair (impartiality): $\overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$
- strongly A-fair (compassion): $\overset{\infty}{\exists} j \geq 0 : A \cap Act(s_j) \neq \emptyset \Rightarrow \overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$
  enabled$(A) \cap$ enabled$(s_j) \neq \emptyset$ taken$(s_j) \in$ taken(A)
- weakly A-fair (justice): $\overset{\infty}{\forall} j \geq 0 : A \cap Act(s_j) \neq \emptyset \Rightarrow \overset{\infty}{\exists} j \geq 0 : \alpha_j \in A$.
  enabled$(A) \cap$ enabled$(s_j) \neq \emptyset$ taken$(s_j) \in$ taken(A)

Checking liveness properties is often done by restricting to fair executions:

$$TS \models_{fair} P \iff FairTraces(TS) \subseteq P$$





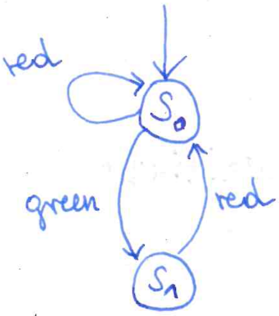
th pay beer b th pay beer ... (soda, soda)

is not unconditionally {soda}-fair

is not strongly {soda}-fair

is weakly {soda}-fair.

Exercise 3.1,
Exercise 3.5,
Exercise 3.6,



execution

s_0 red s_0 red s_0 red s_0 ...

is not weakly {green}-fair.