Lecture 5	02-02-2026

Weak Until, Positive Normal Form   Ex. release

Fairness Conditions in LTL

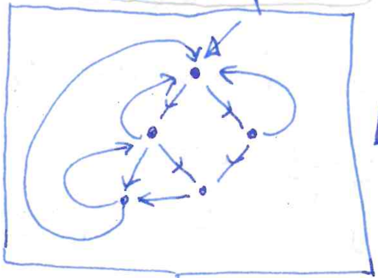
Büchi automata accept ω-regular languages } preparations

From LTL-formulas to Büchi automata

LTL-model-checking algorithm } tomorrow

idea

example



LTL

$\Phi_1 \equiv \Box\Diamond c \vee \Diamond\neg d$

$\Phi_2 \equiv a \, U \circ b$

...

temporal formulas

$\Phi = \Phi_1 \wedge \Phi_2 \wedge \ldots$

MC

✓ $TS \models \Phi$

✗ false

counter

$TS \not\models \Phi$

state

$I \ni s \not\models \Phi$

$\pi \not\models \Phi$

path

$L(\pi) \models \Phi$

Weak Until W

$A_0A_1\ldots = \sigma \models \varphi W \psi :\Leftrightarrow \exists i \geq 0 : [\sigma^{\geq i} \models \psi$ and $\forall 0 \leq j < i : \sigma^{\geq j} \models \varphi]$

OR $\forall i \geq 0 : \sigma_{\geq i} \models \varphi \wedge \neg\psi$

$$\varphi W \psi := (\varphi U \psi) \vee \Box(\varphi \wedge \neg\psi)$$
$$\equiv (\varphi U \psi) \vee \Box\varphi$$

that is: W can be defined from U since $\Box\chi := \neg\Diamond\neg\chi$, $\Diamond\chi := \top U \chi$, forall χ

Then: $\neg(\varphi U \psi) \equiv (\varphi \wedge \neg\psi) U (\neg\varphi \wedge \neg\psi) \vee \Box(\varphi \wedge \neg\psi)$
$\equiv (\varphi \wedge \neg\psi) W (\neg\varphi \wedge \neg\psi)$

Similarly: $\neg(\varphi W \psi) \equiv (\varphi \wedge \neg\psi) U (\neg\varphi \wedge \neg\psi)$

Positive Normal Form of LTL-formulas

$$\varphi ::= \text{true} \mid \text{false} \mid \underset{\in AP}{a} \mid \neg \underset{\in AP}{a} \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \circ\varphi \mid \varphi_1 U \varphi_2 \mid \varphi_1 W \varphi_2$$

Thm. Every LTL-formula is equivalent to an LTL-form. in positive normal form.

Term: $\underbrace{\Box\varphi}_{} \equiv \varphi \; W \; \text{false}$ | compare with $\Diamond\chi := \text{true}\; U\, \chi$

$\overset{def}{=} \neg\Diamond\neg\varphi$

$\overset{def}{\underset{\Diamond}{=}} \neg(\text{true}\; U \neg\varphi)$

$\Diamond\chi := \text{true}\; U\, \chi$

Exercise. Define the release operator $\varphi_1 R \varphi_2$

$\varphi_1 R \varphi_2$: φ_2 must hold for as long as φ_1 is false and <u>also</u> for the first time point in which φ_1 is true.

---

fairness constraints

- unconditional
  $ufair = \Box\Diamond\Psi$
- strong
  $sfair = \Box\Diamond\Phi \rightarrow \Box\Diamond\Psi$
- weak
  $wfair = \Diamond\Box\Phi \rightarrow \Box\Diamond\Psi$

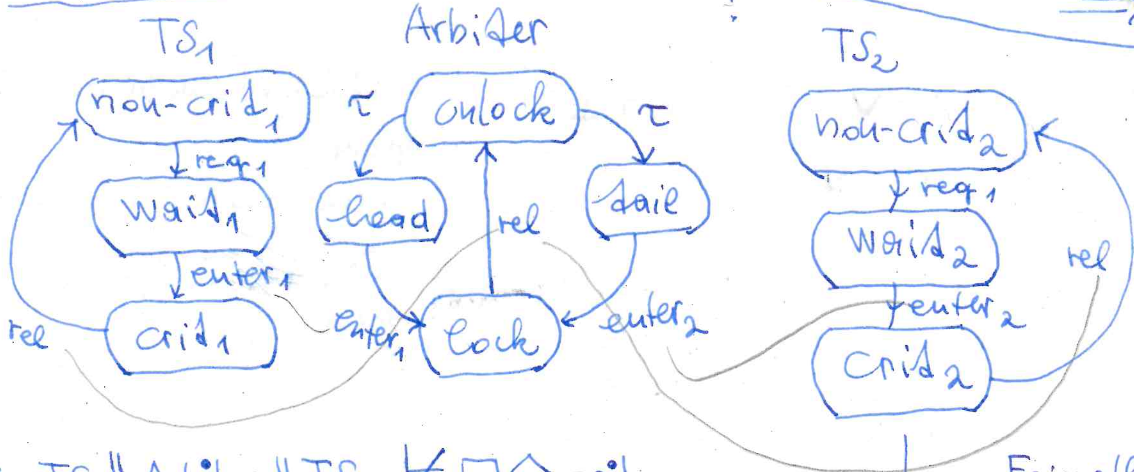
LTS $\mathcal{L} = (S, Act, \rightarrow, I, AP, L)$

$A \subseteq Act$. $\rho: s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \dots$

is unconditionally A-fair
if $\overset{\infty}{\exists} j \geq 0.\ \alpha_j \in A$

strongly A-fair
if $\overset{\infty}{\exists} j \geq 0\ A \cap Act(s_j) \neq \emptyset$
$\Longrightarrow \overset{\infty}{\exists} j \geq 0.\ \alpha_j \in A$

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



Lemma.
$TS \models_{fair} \varphi$
iff $TS \models fair \rightarrow \varphi$

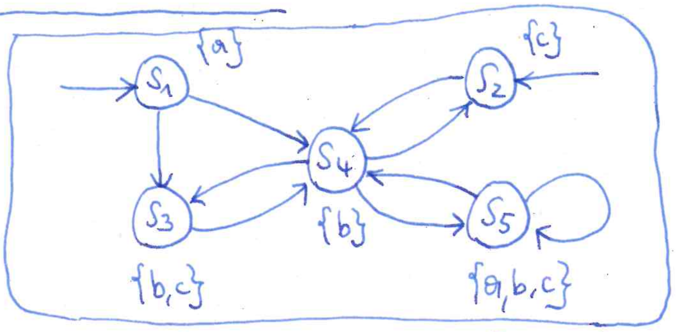
1. $TS_1 \| Arbiter \| TS_2 \not\models \Box\Diamond crit_1$
2. $fair_1 := \Box\Diamond head$
   $TS \| Arbiter \| TS_2 \models_{fair_1} \Box\Diamond crit_1$
3. $fair = (\Box\Diamond head) \wedge (\Box\Diamond tail)$
   $TS_1 \| Arbiter \| TS_2 \models_{fair} (\Box\Diamond crit_1) \wedge (\Box\Diamond crit_2)$
   $\models fair \rightarrow \Box\Diamond crit_1 \wedge \Box\Diamond crit_2$

$Fairpaths(s) := \{\pi \in Paths(s) \,/\, \pi \models fair\}$

$s \models_{fair} \varphi \;:\Leftrightarrow\; \forall \pi \in Fairpaths(s)\ \pi \models \varphi$

$TS \models_{fair} \varphi \Leftrightarrow \forall s \in I_0.\ s_0 \models \varphi$

Exercise 5.2 Consider the transition system over $AP = \{a, b, c\}$:



$TS = \langle \{s_1, \dots, s_5\}, \{a,b,c\}, \rightarrow, \{s_1, s_2\}, \{a,b,c\}, L \rangle$

Decide for each LTL-formula φ_i below, whether $TS \models \varphi_i$ holds. Justify your answers. If $TS \not\models \varphi_i$, provide a path $\pi \in Paths(TS)$ such that $\pi \not\models \varphi_i$.

$\varphi_1 := \Diamond \Box c$ $\quad TS \not\models \varphi_1$, e.g. $s_1(s_3 s_4)^\omega \not\models \Diamond \Box c$

"from some moment on, c holds forever"

$\varphi_2 := \Box \Diamond c$ $\quad TS \models \varphi_2$ ("all infinite paths in TS encounter c infinitely often")

"for infinitely many times, on the path, c holds"

$\varphi_3 := \bigcirc \neg c \rightarrow \bigcirc \bigcirc c$ $\quad TS \models \varphi_3$

$\varphi_4 := \Box a$ $\quad TS \not\models \varphi_4$, because e.g. $s_2 s_4 \ldots \not\models a$

hence $s_2 s_4 \ldots \not\models \Box a$

$\varphi_5 := a \, U \, \Box(b \vee c)$, $\quad TS \models \varphi_5$

$$\left.\begin{array}{l} s_2, \dots, s_5 \models \Box(b \vee c) \\ s_1 \models a \end{array}\right\} \Rightarrow \left.\begin{array}{l} s_1 \models a \, U \, \Box(b \vee c) \\ s_2 \models a \, U \, \Box(b \vee c) \end{array}\right\} \Rightarrow TS \models \varphi_5$$

$\varphi_6 := (\bigcirc \bigcirc b) \, U \, (b \vee c)$, $\quad TS \not\models \varphi_6$ because $s_1 s_4 s_2 \ldots \not\models \bigcirc \bigcirc b$

$s_1 s_4 s_2 \ldots \not\models b \vee c$

$s_1 s_4 s_2 \ldots \not\models (\bigcirc \bigcirc b) \, U \, (b \vee c)$

$TS = \langle S, Act, \rightarrow, I, AP, L \rangle$ $\quad \varphi \in Form_{LTL}(AP)$

$\pi \in Paths(TS)$: $\quad \pi \models \varphi \; :\Longleftrightarrow \; trace(\pi) \models \varphi$

$s \in S$: $\quad s \models \varphi \; :\Longleftrightarrow \; \forall \pi \in Paths(s): \pi \models \varphi$

$TS \models \varphi \; :\Longleftrightarrow \; \forall s \in I: s \models \varphi$

$\varphi_7 := \Diamond \Box b$ $\quad TS \not\models \varphi_7$ because $s_1(s_4 s_2)^\omega \not\models \Diamond \Box b$

because $s_2 \not\models b$

$\varphi_8 := \Box \Diamond b$ $\quad TS \models \varphi_8$ because all paths (infinite!) visit s_4 infinitely often, where b holds

Automata on Infinite Words

Finite-state automata ~ accept finite words, regular Languages
~ used for checking regular safety properties

here: generalization towards more general LT-properties.
(fairness, liveness)

NBAs = non-deterministic Büchi automata

regular expressions: $e ::= \varepsilon \mid a \mid e+e \mid e.e \mid e^*$

~~ε-free~~ 1-free regular expr's: $f ::= a \mid f+f \mid f.f \mid (f^*).f$

ω-regular expressions: $E ::= e.(f)^\omega \mid E+E$

<u>Proposition.</u> $E = e_1 . f_1^\omega + \ldots + e_n . f_n^\omega \Leftrightarrow E$ is a ω-regular expression.

$$\mathcal{L}_\omega(E) = \mathcal{L}(e_1).(\mathcal{L}(f_1))^\omega \cup \ldots \cup \mathcal{L}(e_n).(\mathcal{L}(f_n))^\omega$$

<u>Definition:</u> $\mathcal{L} \subseteq \Sigma^\omega$ Language is ω-regular if $\mathcal{L} = \mathcal{L}_\omega(G)$ for some ω-regular expression G.

$P \subseteq (2^{AP})^\omega$ is ω-regular if P is an ω-regular language over 2^{AP}.
LT-property

NBA $\mathcal{A} = \langle Q, \Sigma, \delta, Q_0, F \rangle$

- Q: finite set of states
- Σ: alphabet
- $\delta: Q \times \Sigma \to 2^Q$
- $Q_0 \subseteq Q$ initial states
- $F \subseteq Q$ acceptance set (accept states)

Thm. An ω language $\mathcal{L} \subseteq (2^{AP})^\omega$ is ω-regular <u>iff</u> it is accepted by a Büchi automaton.

A run for infinite input word $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ is an infinite sequence of states $q_0 q_1 q_2 \ldots$ in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geq 0$.

size $|\mathcal{A}| := |Q| + \bigcup_{\langle q, A \rangle \in Q \times A} |\delta(q, A)|$.

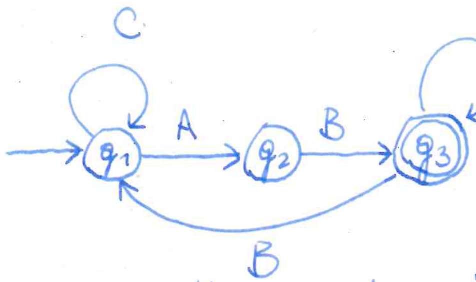
We write $q \xrightarrow{A} p$ if $p \in \delta(q, A)$.

Run $q_0 q_1 q_2 \ldots$ is accepting if $q_i \in F$ for infinitely many $i \geq 0$.

$\mathcal{L}_\omega(\mathcal{A}) = \{ \sigma \in \Sigma^\omega \mid$ there is an accepting run of $\mathcal{A}$ on $\sigma \}$

Example.

$\Sigma = \{A, B, C\}$



$\mathcal{A}$

C^ω has run $q_1 q_1 \cdots = q_1^\omega$. That run is not accepting.

AB^ω has run $q_1 q_2 q_3^\omega$, which is accepting.

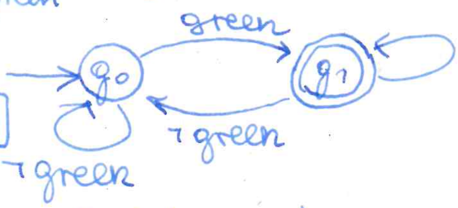
$\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega\left(C^*AB(B^+ + BC^*AB)^\omega\right)$

Liveness property

Example. $AP = \{green, red\}$ "infinitely often green"

$P_{\infty\text{-green}} := \{A_0A_1A_2\ldots \mid \overset{\infty}{\exists} j \geq 0.\ green \in A_j\}$ is accepted by

$\mathcal{A}_{\infty\text{-green}}$



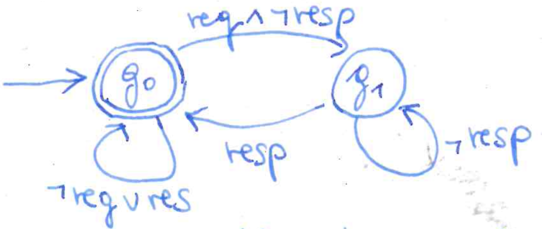
$\sigma = \{green\}\{\}\{green\}\{\}\ldots$ has run $q_0q_1q_0q_1q_0 \cdots = (q_0q_1)^\omega$ which is accepting.

$P_{\infty\text{-green}} = \mathcal{L}_\omega(\mathcal{A}_{\infty\text{-green}})$ $\sigma' = (\{green, red\}\{\}\{green\}\{red\})^\omega$ has the same accepting run (possibly instantly)

Example. "Whenever there is a request, eventually there is a response."

$P_{req/res} := \{A_0A_1A_2\ldots \in AP^\omega \mid \forall i \geq 0.[req \in A_i \Rightarrow \exists j \geq i.\ res \in A_j]\}$

$AP = \{req, res\}$



$\mathcal{A}_{req/res}$

$2^{AP} \setminus \mathcal{L}_\omega(\mathcal{A}_{req/res}) = \{A_0A_1\ldots \in (2^{AP})^\omega \mid \exists i \geq 0.(req \in A_i \land \forall j \geq i.(res \notin A_j))\}$

Liveness property

"Some request is never answered by a response"

$\mathcal{L}_\omega(P_{\infty\text{-green}}) = (\{green, red\}^*\ green)^\omega$

$\mathcal{L}_\omega(P_{req/res}) = \left(\emptyset + \{res\} + \{req, res\} + \{req\}.(\emptyset + \{req\})^*.(\{res\} + \{res, req\})\right)^\omega$

Regular Safety Properties

A safety property P_{safe} is <u>regular</u> if its set of bad prefixes is a regular language over 2^{AP}.

<u>E.g.</u> Every invariant over AP is regular:

$$P = \{A_0A_1A_2\ldots \in (2^{AP})^\omega \mid A_i \models \Phi \text{ for all } i \geq 0\} \text{ for some formula } \Phi$$

Its bad prefixes are:

$$BP(P) = (\{A \in 2^{AP} \mid A \models \Phi\})^* . \{A \in 2^{AP} \mid A \not\models \Phi\} . (2^{AP})^*$$

$$\sim \quad \Phi^* \quad . \quad (\neg\Phi) \quad . \quad true^*$$

Concretely: $AP = \{a, b\}$, $\Phi_0 = a \vee \neg b$ $\qquad P_0 := \{\sigma \in (2^{AP})^\omega \mid \sigma \models \Phi\}$

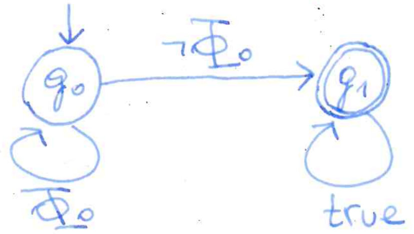
$\Phi_0 \sim \{\emptyset, \{a\}, \{a,b\}\}$

$\neg\Phi_0 \sim \{\{b\}\}$

$true \sim 2^{AP} = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$

$$BP(P_0) = (\{\emptyset, \{a\}, \{a,b\}\})^* . \{b\} . (\{\emptyset, \{a\}, \{b\}, \{a,b\}\})^*$$

$$\sim (\Phi_0)^* . (\neg\Phi_0) . (true)^*$$



DFA for $BP(P_0)$

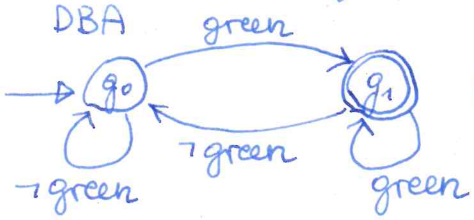
LTL-formulas $\longrightarrow$ Büchi automata

Examples (Motivation).

(i) NBA for $\Box\Diamond$ green "infinitely often green"

deterministic!

DBA



$\boxed{\mathcal{A}_1}$

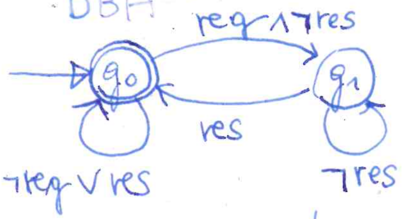
$AP \supseteq \{green\}$

$\mathcal{L}_\omega(\mathcal{A}_1) =$
$= \{A_0A_1A_2\ldots \in 2^{AP} \,/\, \overset{\infty}{\exists} j \geq 0: green \in A_j\}$
$=$ Words($\Box\Diamond$ green)

(ii) NBA for $\Box$(request $\longrightarrow \Diamond$ response)

$AP \supseteq \{req, res\}$

DBA

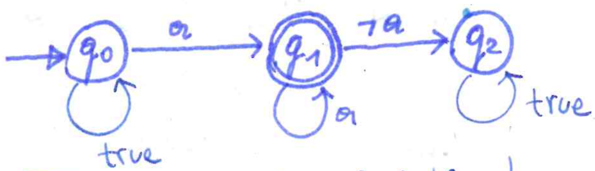


$\boxed{\mathcal{A}_2}$

deterministic!

Words($\Box$(req $\rightarrow \Diamond$ res)) =
$= \{A_0A_1A_2\ldots \in (2^{AP})^\omega \,/\, \forall j \geq 0\, [req \in A_j \Rightarrow$
$\Rightarrow \exists k \geq j: res \in A_k]\}$
$= \{A_0A_1A_2\ldots \in (2^{AP})^\omega \,/\, (\neg \overset{\infty}{\exists} j \geq 0.\ req \in A_j)$
$\vee (\overset{\infty}{\exists} j \geq 0.\ res \in A_j)\}$
$= \mathcal{L}_\omega(\mathcal{A}_2)$.

(iii) NBA for $\Diamond\Box a$. "eventually always a"



$\boxed{\mathcal{A}_3}$ non-deterministic!

Words($\Diamond\Box a$) =
$= \{A_0A_1A_2\ldots \in (2^{AP})^\omega \,/\, \overset{\infty}{\forall} j \geq 0: a \in A_j\}$
$= \{\ldots \,/\, \exists k \geq 0\ \forall j \geq k: a \in A_j\}$
$= \mathcal{L}_\omega(\mathcal{A}_3)\}$

(ii)' NBA for "every continued request a response must follow (perhaps immediately)"

$\Box$(req $\rightarrow \Diamond(\neg$req $\vee$ res)] $\equiv \Box$ (req $\rightarrow$ T U($\neg$req $\vee$ res))

From the lecture, Thor's (improved) idea:
$\Box$(req $\rightarrow$ req U($\neg$req $\vee$ res))
are equivalent

Julius Richard Büchi (1924–1984)

Büchi automaton

NBA $\mathcal{A} = \langle Q, \Sigma, \delta, Q_0, F\rangle$

Q: finite set of states

$\underbrace{2^{AP}}_{\text{in our case subsets of prop. variables}} = \Sigma$: alphabet

because we need to accept traces

$\delta: Q \times \Sigma \rightarrow 2^Q$ $\quad$ $\delta: Q \rightarrow 2^{Q\times\Sigma}$ coalgebraic formulation

$Q_0 \subseteq Q$ initial states

$F \subseteq Q$ final states.

We write $q \xrightarrow{A} q'$ for $q' \in \delta(q, A)$

A run for infinite input word $\sigma = A_0A_1A_2\cdots \in \Sigma^\omega$ is an infinite sequence of states $Q_0 \ni q_0q_1q_2\ldots \in Q^\omega$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geq 0$.

Run $q_0q_1q_2\ldots$ is accepting if $q_i \in F$ for infinitely many $i \geq 0$

$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega \,/\,$ there is an accepting run of $\mathcal{A}$ on $\sigma\}$