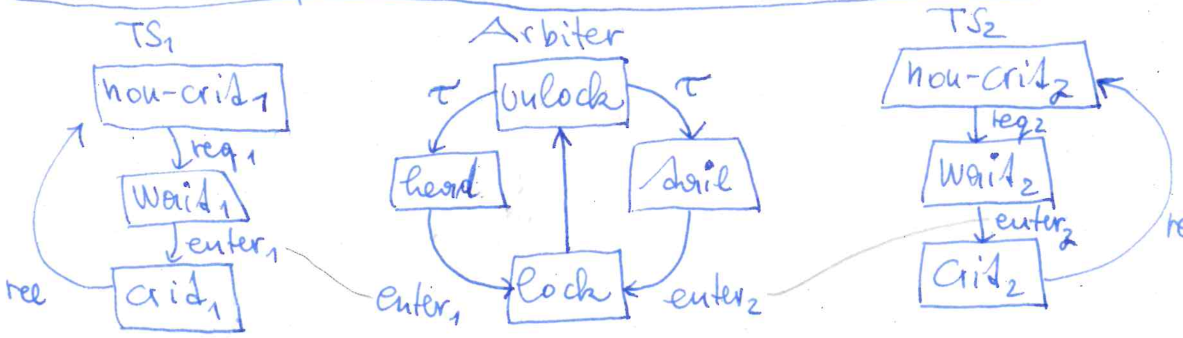
Lecture 5 | 07-02-2025

Fairness
Weak Until, Positive Normal Form — Ex. release.
Büchi automata accept ω-regular languages
From LTL-formulas to Büchi automata
LTL-model-checking algorithm
- idea
- example

fairness constraints
- unconditional: $ufair = \Box\Diamond\psi$
- strong: $sfair = \Box\Diamond\Phi \to \Box\Diamond\Psi$
- weak: $ufair = \Diamond\Box\Phi \to \Box\Diamond\Psi$



$TS_1 \| Arbiter \| TS_2 \not\models \Box\Diamond crit_1$

$fair_1 = \Box\Diamond head$

$TS_1 \| Arbiter \| TS_2 \models_{fair_1} \Box\Diamond crit_1$

$fair = \Box\Diamond head \land \Box\Diamond tail$

$TS_1 \| Arbiter \| TS_2 \models_{fair} \Box\Diamond crit_1 \land \Box\Diamond crit_2$

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

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

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

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

$\models fair \to \Box\Diamond crit_1 \land \Box\Diamond crit_2$

Weak until

$\sigma \models \varphi W \psi :\Leftrightarrow (\exists i \geq 0: \sigma^{\geq i} \models \psi$ and $\forall 0 \leq j < i: \sigma^{\geq j} \models \varphi)$
OR $\forall i \geq 0: \sigma_{\geq i} \models \varphi \land \neg\psi.$

$\Rightarrow \quad \varphi W \psi := \varphi U \psi \lor \Box(\varphi \land \neg\psi)$
$\equiv \varphi U \psi \lor \Box\varphi$

Then: $\neg(\varphi U \psi) \equiv (\varphi \land \neg\psi) U (\neg\varphi \land \neg\psi) \lor \Box(\varphi \land \neg\psi)$
$\equiv (\varphi \land \neg\psi) W (\neg\varphi \land \neg\psi)$

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

Positive Normal Form of LTL-formulas

$\varphi ::= true \mid false \mid \underset{\in AP}{a} \mid \neg \underset{\in AP}{a} \mid \varphi_1 \land \varphi_2 \mid \varphi_1 \lor \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 U \varphi_2 \mid \varphi_1 W \varphi_2$

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

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

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

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: $E ::= e.(f)^\omega \mid E+E$

Proposition. $E = e_1.f_1^\omega + \ldots + e_n.f_n^\omega \iff E$ is a ω-regular expression.

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

Definition: $\mathcal{L} \subseteq \Sigma^\omega$ 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

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

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)

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

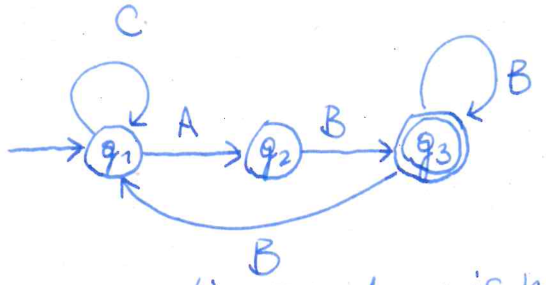
We write $q \xrightarrow{A} p$ if $p \in \delta(q, A)$.

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.

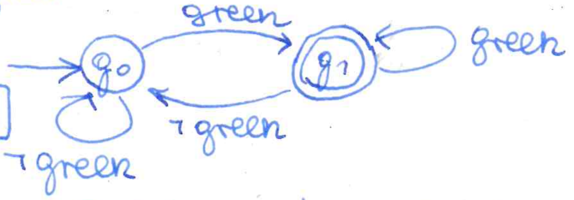
$(AB)^\omega$ has run $q_1 q_2 q_3^\omega$, which is accepting.

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

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

$P_{\infty\text{-green}} := \{ A_0 A_1 A_2 \dots \mid \overset{\infty}{\exists} j \geq 0.\ green \in A_j \}$ is accepted by

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



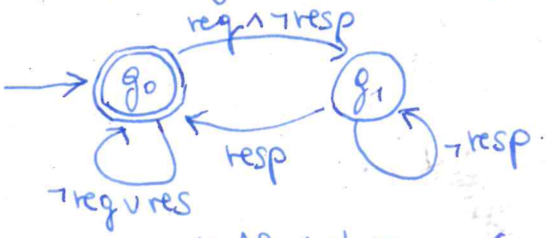
$\sigma = \{green\}\{\}\{green\}\{\} \dots$ has run $q_0 q_1 q_0 q_1 q_0 \dots = (q_0 q_1)^\omega$ which is accepting.

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

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

$P_{req/res} := \{ A_0 A_1 A_2 \dots \in AP^\omega / \overset{\infty}{\exists} j \geq 0.\ res \in A_j \}$

$AP = \{req, res\}$



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

$$2^{AP} \setminus \mathcal{L}_\omega(\mathcal{A}_{req/res}) = \{ A_0 A_1 \dots \in (2^{AP})^\omega \mid \exists i \geq 0. (req \in A_i \wedge \forall j \geq i. (res \notin A_j)) \}$$

"Some request is never answered by a response"

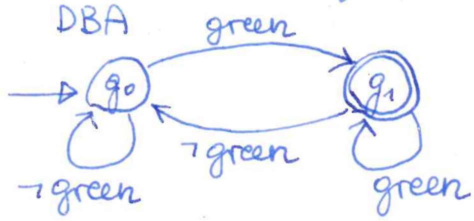
$$\mathcal{L}_\omega(P_{\infty\text{-green}}) = (\{green, red\}^* green)^\omega$$

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

LTL-formulas → Büchi automata

Examples (Motivation). (i) NBA for $\Box\Diamond$ green ("infinitely often green") — *deterministic!*

DBA



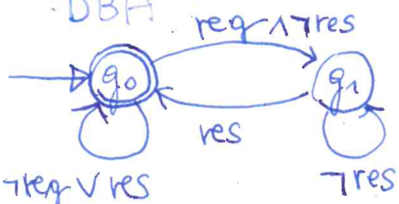
$\mathcal{A}_1$

$AP \supseteq \{green\}$

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

(ii) NBA for $\Box$(request → $\Diamond$ response) $AP \supseteq \{req, res\}$

DBA

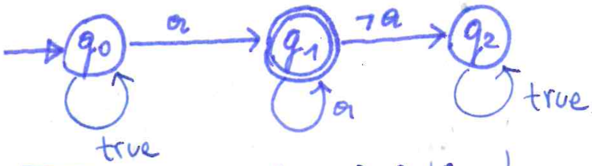


$\mathcal{A}_2$

deterministic!

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

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

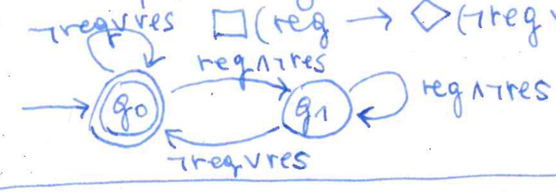


$\mathcal{A}_3$ *non-deterministic!*

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)"

$\Box(req \to \Diamond(\neg req \vee res)] = \Box(req \to T\,U(\neg req \vee res))$



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

$2^{AP} = \Sigma$: alphabet (in our case subsets of prop. variables because we need to accept traces)

$\delta : Q \times \Sigma \to 2^Q$ ≅ $\delta : Q \to 2^{Q \times \Sigma}$ 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 <u>run</u> for infinite input word $\sigma = A_0A_1A_2\ldots \in \Sigma^\omega$ 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 <u>accepting</u> if $q_i \in F$ <u>for infinitely many</u> $i \geq 0$

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

φ an LTL-formula.

G_φ is constructed such that it accepts all words $\sigma = A_0 A_1 A_2 \dots \in Word(\varphi)$.

Hereby a word $\sigma = A_0 A_1 A_2 \dots$ will be accepted

by a run $\bar{\sigma} = B_0 B_1 B_2 \dots$ with states $B_i \supseteq A_i$

where B_i a maximal subset of subformulas

or negated subformulas of φ

such that: $\psi \in B_i \iff A_i A_{i+1} A_{i+2} \dots \models \psi$

<u>Ex.</u> $\varphi = a \, U (\neg a \wedge b)$ $\quad \sigma = \{a\}\{a,b\}\{b\}\dots$

$$B_i \subseteq \underbrace{\{a, b, \neg a, \neg a \wedge b, \varphi\}}_{\text{subformulas of } \varphi} \cup \underbrace{\{\neg a, \neg b, \neg(\neg a \wedge b), \neg\varphi\}}_{\text{negated subformulas of } \varphi}$$

$B_0 = \{a, \neg b, \neg(\neg a \wedge b), \varphi\}$

$B_1 = \{a, b, \neg(\neg a \wedge b), \varphi\}$

$B_3 = \{b, \neg a, \neg a \wedge b, \varphi\}$

The semantics of the <u>next-step operator</u> relies on a non-local condition and will be encoded in the transition relation.

The meaning of the until-operator is split according to <u>the expansion law</u> into local conditions (encoded in the states)

$(\varphi_1 U \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 U \varphi_2))$ and a next-step condition (encoded in transitions).

Closure of an LTL-formula φ:

closure(φ) := $\{\psi \mid \psi$ is subformula of φ, or a negation of a subformula of $\varphi\}$
where we identify $\neg\neg\psi$ with ψ

example: closure$(\underbrace{a\, U (\neg a \wedge b)}_{=:\varphi}) = \{a, b, \neg a, \neg b, \neg a \wedge b, \neg(\neg a \wedge b), \varphi, \neg\varphi\}$

$|\text{closure}(\varphi)| \in O(|\varphi|)$

Elementary Sets of formulas:

$B \subseteq$ closure(φ) is elementary if B is consistent w.r.t. prop. logic,
maximal,
locally consistent w.r.t. Until

B is consistent w.r.t. prop. logic:

$$\left.\begin{array}{l} \varphi_1 \wedge \varphi_2 \in B \iff \varphi_1 \in B \text{ and } \varphi_2 \in B \\ \psi \in B \implies \neg\psi \notin B \\ \text{true} \in \text{closure}(\varphi) \implies \text{true} \in B \end{array}\right\} \text{ for all } \varphi_1 \wedge \varphi_2,\ \psi \in B$$

B is locally consistent w.r.t. Until-operator:

$$\left.\begin{array}{l} \varphi_2 \in B \implies \varphi_1 U \varphi_2 \in B \\ \varphi_1 U \varphi_2 \in B \text{ and } \varphi_2 \notin B \implies \varphi_1 \in B \end{array}\right\} \text{ for all } \varphi_1 U \varphi_2 \in \text{closure}(\varphi)$$

B is maximal:

$$\psi \notin B \implies \neg\psi \in B \quad \} \text{ for all } \psi \in \text{closure}(\varphi).$$

Local consistency condition for the until operator is due to:

$$\varphi_1 U \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 U \varphi_2)).$$

Maximality and consistency imply:

$$\psi \in B \iff \neg\psi \notin B.$$

$$\varphi_1, \varphi_2 \notin B \implies \varphi_1 U \varphi_2 \notin B$$

Thm. For every LTL-formula φ over AP there exists a GNBA G_φ over 2^{AP} such that
(a) $\text{Words}(\varphi) = \mathcal{L}_\omega(G_\varphi)$
(b) G_φ can be constructed in time $2^{O(|\varphi|)}$
(c) The number of accepting sets of G_φ is bounded above by $O(|\varphi|)$.

Proof. $G_\varphi = (Q, 2^{AP}, \delta, Q_0, \mathcal{F})$

where $Q := \{ B \subseteq \text{closure}(\varphi) \,/\, B \text{ is elementary} \}$

$Q_0 := \{ B \in Q \,/\, \varphi \in B \}$

$\mathcal{F} := \{ F_{\varphi_1 U \varphi_2} \,/\, \varphi_1 U \varphi_2 \in \text{closure}(\varphi) \}$

where $F_{\varphi_1 U \varphi_2} = \{ B \in Q \,/\, \varphi_1 U \varphi_2 \notin B \text{ or } \varphi_2 \in B \}$

$\delta : Q \times 2^{AP} \to 2^Q$

$$\langle B, A \rangle \longmapsto \delta(B,A) := \begin{cases} \emptyset & \ldots \text{ if } A \neq B \cap AP \\ \{B' \ldots & \text{if } A = B \cap AP \\ \text{where } B' \text{ is elementary such that} & \\ \text{(i) } (\bigcirc\psi \in B \Leftrightarrow \psi \in B') \text{ for all } \bigcirc\psi \in \text{closure}(\varphi) & \\ \text{(ii) for every } \varphi_1 U \varphi_2 \in \text{closure}(\varphi) & \\ \quad \varphi_1 U \varphi_2 \in B \Updownarrow \varphi_2 \in B \vee (\varphi_1 \in B \wedge \varphi_1 U \varphi_2 \in B') & (*) \end{cases}$$

$= \{ B' \in Q \,/\, (*) \}$

$$\varphi_1 U \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 U \varphi_2))$$

For the definition of $F_{\varphi_1 U \varphi_2} = \{ B \in Q \,|\, \varphi_1 U \varphi_2 \notin B \text{ or } \varphi_2 \in B \}$ note:

$$\overset{\infty}{\exists} j \geq 0 : B_j \in \underbrace{F_{\varphi_1 U \varphi_2}}_{= \{ B \in Q / \varphi_1 U \varphi_2 \notin B \text{ or } \varphi_2 \in B \}} \Longleftrightarrow \neg \overset{\infty}{\forall} j \geq 0 : B_j \in \underbrace{Q \setminus F_{\varphi_1 U \varphi_2}}_{= \{ B \in Q / \varphi_1 U \varphi_2 \in B \text{ and } \varphi_2 \notin B \}}$$

Example. GNBA for $\varphi = \bigcirc a$ $\quad AP = \{a\}$.

$\mathcal{G}_{\bigcirc a} = \langle Q, 2^{AP}, \delta, Q_0, \mathcal{F} \rangle$

$closure(\varphi) = \{a, \bigcirc a, \neg a, \neg\bigcirc a\}$

$Q = \{B \subseteq closure(\varphi) \mid B \text{ is elementary}\} = \{B_1, B_2, B_3, B_4\}$

$B_1 = \{a, \bigcirc a\}$, $B_2 = \{a, \neg\bigcirc a\}$, $B_3 = \{\neg a, \bigcirc a\}$, $B_4 = \{\neg a, \neg\bigcirc a\}\}$

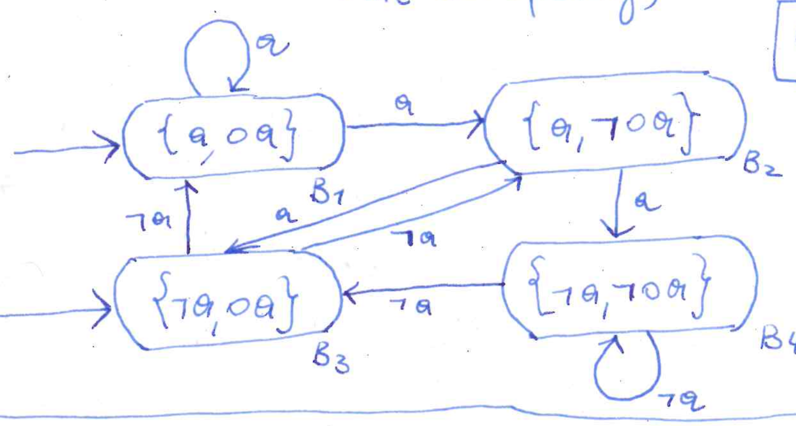
$Q_0 = \{B \in Q \mid \varphi \in B\} = \{B_1, B_3\}$

$\mathcal{F} = \{F_{\varphi_1 U \varphi_2} \mid \varphi_1 U \varphi_2 \in closure(\varphi)\} = \emptyset$ (hence all infinite traces are accepting)

$\delta : Q \times 2^{AP} \to 2^Q$

$\langle q, A \rangle \mapsto \delta(q, A)$

$\boxed{\mathcal{G}_{\bigcirc a}}$



e.g.

$\delta(B_2, \{a\}) =$

$= \{B_2' \in Q \mid (\bigcirc\psi \in B_2 \Leftrightarrow \psi \in B_2')$ for all $\bigcirc\psi \in cl(\varphi)\}$

$= \{B_2' \in Q \mid \bigcirc a \in B_2 \Leftrightarrow a \in B_2'\}$

$= \{B_2' \in Q \mid \cancel{a} \notin B_2'\}$

$= \{B_2' \in Q \mid \neg a \in B_2'\}$

$= \{B_3, B_4\}$

$\sigma = \neg a \; a \; \neg a \; a \; \neg a \; \neg a \; a \; a \; a \; \neg a \; \ldots \in AP^\omega$

$\bar{\sigma} = B_3 \, B_2 \, B_3 \, B_2 \, B_4 \, B_3 \, B_1 \, B_1 \, B_2 \; \ldots \in Q^\omega$

$Words(\overset{\varphi}{\bigcirc a}) \overset{?}{=} \mathcal{L}_\omega(\mathcal{G}_{\bigcirc a})$ $\quad$ We have to check:

$\sigma = A_0 A_1 A_2 \ldots \models \bigcirc a \quad \overset{?\checkmark}{\Longleftrightarrow} \quad \bar{\sigma} := \bar{B}_0 \bar{B}_1 \bar{B}_2 \ldots$ accepting run of $\mathcal{G}_{\bigcirc a}$

where $\bar{B}_i := \{\psi \in Q \mid A_i A_{i+1} \ldots \models \psi\} \in \{B_1, B_2, B_3, B_4\}$

$\Updownarrow \qquad\qquad \Updownarrow$

$A_1 = \{a\}$ $\quad$ second transitions must be a $\quad$ $\bar{\sigma}$ is run of $\mathcal{G}_{\bigcirc a}$

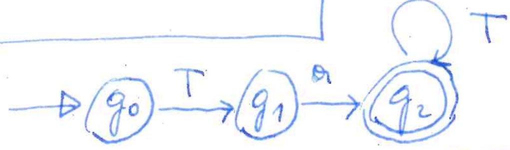
since for all $A_0' A_1' A_2' \ldots \in (2^{AP})^\omega$ it holds:

$\bar{B}_0' \bar{B}_1' \bar{B}_2' \ldots$ is a run of $\mathcal{G}_{\bigcirc a}$ that starts in $\bar{B}_0$ (which does not need to be a start state)

with $\bar{B}_i' := \{\psi \in Q \mid A_i' A_{i+1}' \ldots \models \psi\}$

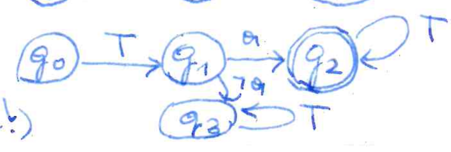
DBA

A 3-state NBA for $\bigcirc a$ is:



incomplete

4-state NBA/DBA is



complete.

(non-blocking!)

Example 2 $\quad \varphi = a\,\mathcal{U}\,b \qquad AP = \{a, b\}$

$closure(\varphi) = \{a, b, \neg a, \neg b, a\,\mathcal{U}\,b, \neg(a\,\mathcal{U}\,b)\}$

$Q := \{B \subseteq closure(\varphi) \,/\, B \text{ is elementary}\} = \{B_1, B_2, B_3, B_4, B_5\}$

$B_1 = \{a, b, \varphi\} \qquad B_4 = \{a, \neg b, \neg\varphi\}$

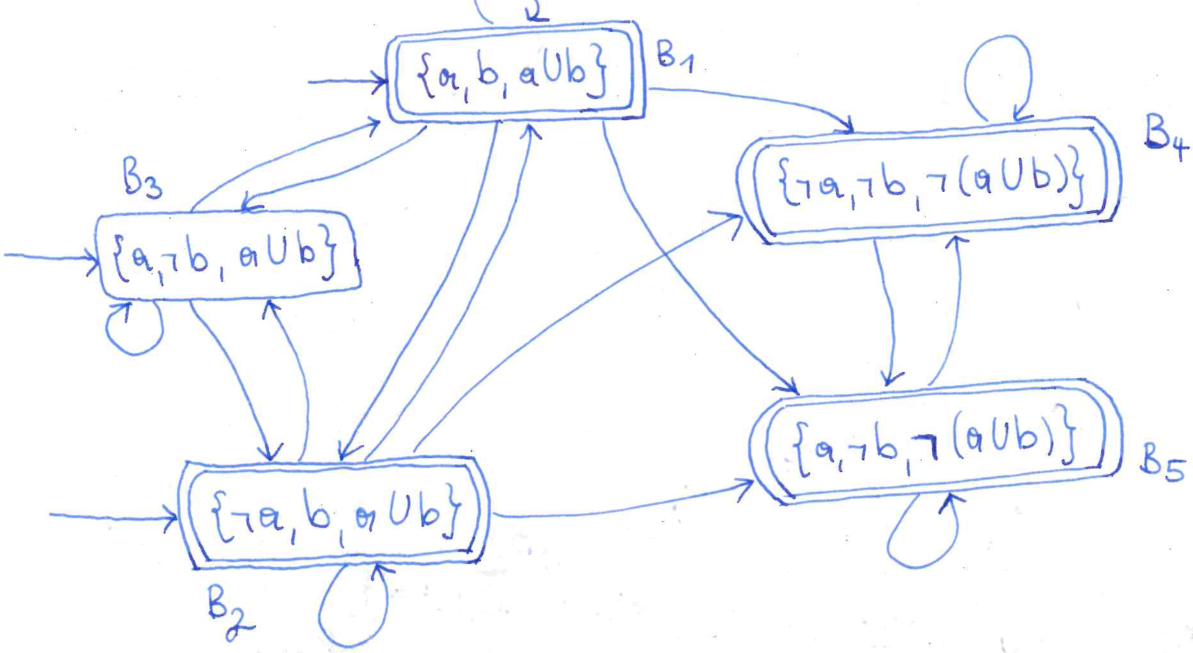
$B_2 = \{\neg a, b, \varphi\} \qquad B_5 = \{\neg a, \neg b, \neg\varphi\}$

$B_3 = \{a, \neg b, \varphi\}$

note that $\{\neg a, \neg b, \varphi\}$ and $\{\neg a, b, \neg\varphi\}$, $\{a, b, \neg\varphi\}$ are <u>not</u> locally consistent, hence not elementary.

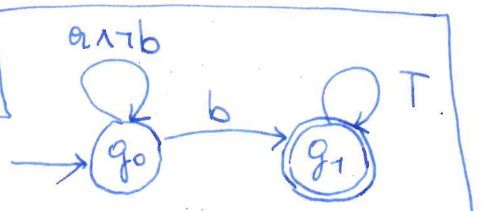
$Q_0 := \{B \in Q \mid \varphi = a\,\mathcal{U}\,b \in B\} = \{B_1, B_2, B_3\}$

$\mathcal{F} := \{F_{a\mathcal{U}b}\}$ where $F_{a\mathcal{U}b} = \{B \in Q \mid a\,\mathcal{U}\,b \notin B \text{ or } b \in B\} = \{B_1, B_2, B_4, B_5\}$



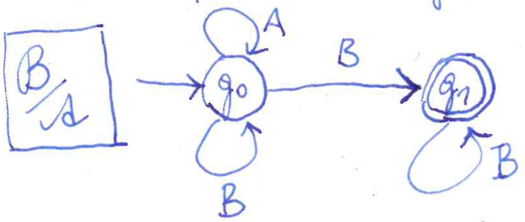
DBA

A 2-state NBA for $\varphi = a\,\mathcal{U}\,b$ is:



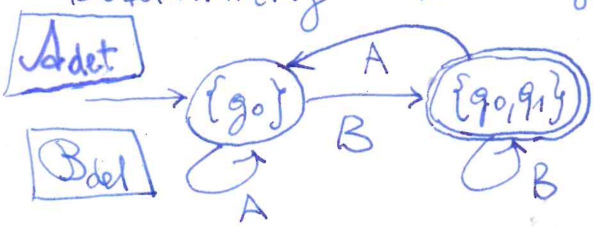
NBAs are more powerful than DBAs

NBA $\mathcal{B}$ for the ω-regular expression $(A+B)^* B^\omega$:



(as $\boxed{\text{NFA}}$ $\mathcal{A}$ its language is described by the regular expression $(A+B)^* . B . B^*$)

Determinizing NFA $\mathcal{A}$ by the subset construction yields:



its Language is: $A^*B.(B^* + AA^*B)^*$

NOTE that: $\mathcal{L}(\mathcal{A}det) = \mathcal{L}(A^*B.(B^*+AA^*B)^*)$

$\overset{!}{=} \mathcal{L}((A+B)^* B^+)$

$= \mathcal{L}((A+B)^* B B^*) = \mathcal{L}(\mathcal{A})$

But note that:

$$\mathcal{L}_\omega(\mathcal{B}_{det}) = \mathcal{L}_\omega(A^*B.(B^+ + A^+B)^\omega)$$

$$= \{ w \in \{A,B\}^\omega \mid w \text{ contains } \infty\text{-many } B \}$$

$$\neq \{ w \in \{A,B\}^\omega \mid w \text{ contains only finitely many } A \}$$

$$= \mathcal{L}_\omega((A+B)^* B^\omega)$$

$$= \mathcal{L}_\omega(\mathcal{B})$$

This shows: The subset construction for determinizing NFAs to DFAs does not work for determinizing NBAs to DBAs.

$\boxed{\text{Theorem.}}$ There does not exist an NBA such that

$$\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega((A+B)^* B^\omega)$$

Proof.: See proof of Thm 4.50 in the book!