

Lecture 1 (30-07-2021)

30 July 2021 09:28

Automata on infinite words

Some notations

- Let Σ be a finite nonempty set (called alphabet).
- A finite word (over Σ) is a finite sequence w of letters from Σ .
 $w = a_0 a_1 \dots a_n, \quad a_i \in \Sigma.$
- ϵ is the empty word.
- Σ^* is the set of all finite words (over Σ).
- An infinite word over Σ is an infinite sequence of letters from Σ .
 $\alpha = a_0 a_1 a_2 \dots, \quad a_n \in \Sigma \quad \forall n \in \mathbb{N}_0.$

(Different formality: $\mathbb{N}_0 = \{0, 1, \dots\}$ and $\alpha: \mathbb{N}_0 \rightarrow \Sigma^*$)

Also, $\omega = \mathbb{N}_0$.

$\Sigma^\omega =$ all infinite words (on Σ)

(In general, given sets X and Y , Y^X denotes the set of all functions $x \rightarrow y$)

Examples ① $\Sigma = \{a, b\}$

$\alpha = a b a b a b \dots$

or: $\alpha(n) = \begin{cases} a & ; \quad 2|n \\ b & ; \quad 2 \nmid n \end{cases}$

② $\alpha = a b b a b b a b b$

$\alpha(n) = \begin{cases} a & ; \quad 3|n \\ b & ; \quad 3 \nmid n \end{cases}$

$\alpha = (abb)^\omega$

③ $\gamma: \omega \rightarrow \{a, b\}$

$\gamma(n) = \begin{cases} a & ; \quad n \text{ is prime} \\ b & ; \quad \text{otherwise} \end{cases}$

$\gamma = \underset{0}{b} \underset{1}{b} \underset{2}{a} \underset{3}{a} \underset{4}{b} \underset{5}{a} \underset{6}{b} \underset{7}{a} \underset{8}{b} \underset{9}{b} \underset{10}{a} \dots$

γ has infinitely many 'a's and 'b's.
(Can't write as compactly as before.)

$$\textcircled{1} \quad \Sigma = \{a\}, \quad \Sigma^\omega = \{a^\omega\}. \quad (|\Sigma| = 1)$$

But if $|\Sigma| > 1$, then Σ^ω is not a countable set.
OTOH, Σ^* is always a countable set. ($1 \leq |\Sigma| < \infty$)

Automata:

$$A = (Q, \Sigma, q_0, \Delta \subset Q \times \Sigma \times Q, \text{"Acceptance condition"})$$



→ a finite set of states

→ $q_0 \in Q$ → the initial state (unique)

→ $\Delta \subset Q \times \Sigma \times Q$ → the transition relation

$$(q, a, q') \in \Delta \equiv q \xrightarrow{a} q'$$

(Non-determinism here is fine)

Now, let $\alpha = a_0 a_1 a_2 \dots \in \Sigma^\omega$ be given.

A run β of A on α is an infinite sequence of states

$$\beta = q_0 q_1 q_2 \dots$$

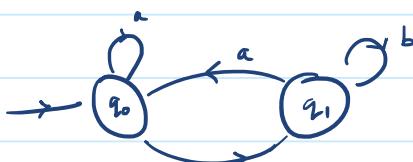
(run)

such that " q_0 " is indeed the initial state and

$$\forall i \in \omega : (q_i, a_i, q_{i+1}) \in \Delta$$

In terms of functions: Given $\alpha : \omega \rightarrow \Sigma$, we have
 $f : \omega \rightarrow \Sigma$ s.t. $f(0) = q_0$ and $(f(n), \alpha(n), f(n+1)) \in \Delta \quad \forall n \in \omega$.

Example.



$$\alpha = (ab)^\omega$$

$$\alpha = a \cdot (a \cdot b)^\omega$$

$$f = q_0 (q_0 q_1)^\omega$$

"Acceptance condition": (Büchi automata)

$\alpha \rightarrow$ input word

$f \rightarrow$ a run of A on α

$\text{Inf}(f) :=$ the set of states which occur infinitely often along f

$$= \{q \in Q : \exists^\infty i \in \omega \text{ s.t. } f(i) = q\}$$

Obs. $\text{Inf}(f) \neq \emptyset$. (There are only finitely many states.)

Büchi automaton (BA): fix $G \subseteq Q$ called the "good state".

A run f is accepted by a BA if $\text{Inf}(f) \cap G \neq \emptyset$.

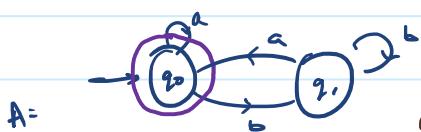
(Thus, some good state appears infinitely often.)

A word $\alpha \in \Sigma^\omega$ is accepted by A if α has an accepting run f on the word α .

$$L(A) := \{ \alpha \in \Sigma^\omega : A \text{ accepts } \alpha \}$$

↳ Language of A

Example



$$G = \{q_0\}, \Sigma = \{a, b\}$$

Claim. $L(A) = \{\alpha \in \Sigma^\omega : \alpha \text{ has inf.}\}$

many 'a' s.

Prof. Let the right side be L.

- $L(A) \subseteq L$:

$$\alpha \in L(A), \quad \alpha = a_0 a_1 a_2 \dots$$

Note that A is deterministic, thus α has a unique run f , which is accepted.

$$f = q_0 q'_1 q'_2 q'_3 \dots$$

Thus, q_0 appears inf. often above. Since it only receives 'a', we see that 'a' appears inf. often.

- $L \subseteq L(A)$:

Let $\alpha \in L$. It has a unique run f .

Then, since α has inf. many 'a's, f will have inf. many ' q'_0 's.

B

Can also write $L = (b^* a)^\omega$ once we have defined what that means.

Question What about $\overline{L} = \Sigma^\omega \setminus L$? Can that be accepted by a Büchi automaton?

Lecture 2 (04-08-2021)

04 August 2021 09:31

Note. We do NOT allow ϵ transitions in this course.

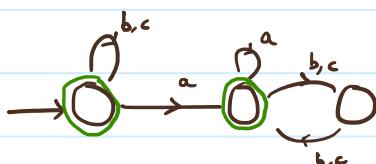
Fact. Even though we insisted on single initial state, the expressive power does not change if we allow more.
(It is simply for convenience.)

Example. (1) L over $\Sigma = \{a, b, c\}$.

$L = \text{every 'a' is eventually followed by a 'b'}$



(2) $L_2 = \text{any two occurrences of 'a' are separated by even no. of other (b, c) letters}$

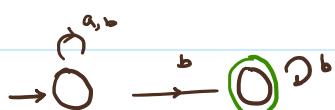


(3) $\Sigma = \{a, b\}$, $L = \text{inf. many 'a's}$

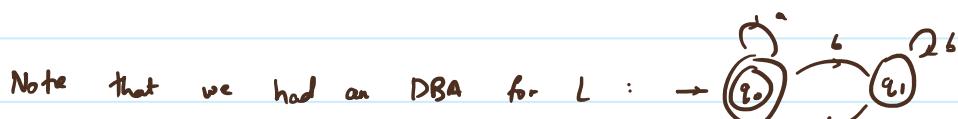


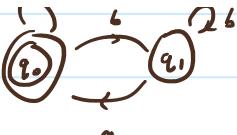
Complement: $\bar{L} = \Sigma^\omega \setminus L = \text{finitely many 'a's}$

Q. What is a BA for \bar{L} ?



Q. Do we have a deterministic Büchi automaton (DBA) for \bar{L} ?



Note that we had an DBA for L : 

$$A : G = \{q_0\}$$

Toggle states . $A' : G = \{q_1\}$

But $L(A') \not\supseteq \bar{L}$.

\downarrow
infinitely
many b

\downarrow
eventually
 a

$$(ab)^\omega \in L(A') \text{ but } (ab)^\omega \notin \bar{L}$$

Complementing the good state of a DBA does

NOT complement the accepted language.

Claim: There is no DBA for $\bar{L} = \{\alpha \in \Sigma^\omega : \alpha \text{ has finitely many } 'a's\}$.

Thus, as opposed to finite languages, non-determinism actually gives us more languages.

Proof. We prove this by contradiction.

For the sake of contradiction, assume that \exists DBA A such that $L(A) = \bar{L}$.

Suppose A has m states.

$$\alpha_0 = b^\omega = b\ b b \dots \in \bar{L}$$

$$f_0 = q_0\ q_1\ q_2 \dots$$

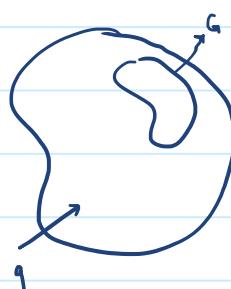
↳ unique run of α_0 .

Since f_0 is accepting, $\exists n_1 \text{ s.t. } q_{n_1, n_1} \in G$.

$\underbrace{b\ b \dots b}_{n_1} \mid b\ b \dots$ Pick the smallest such n_1 .

$$f_0 = \boxed{\quad} \uparrow \text{first good state}$$

Define $\alpha_1 := b^{n_1} a b^\omega \in \bar{L}$.



$\alpha_1 = b \cdots b^{n_1} a b b \cdots$
 $\beta_1 = \square \cdots \square \xrightarrow{\text{EG}} \cdots \square$
 \nwarrow_{EG} again a
good state

Then, we can get n_2 s.t. $b^{n_1} a b^{n_2} a$ ends at a good state.

Then, $\alpha_2 = b^{n_1} a b^{n_2} a b^\omega \in L$.

Its unique run β_2 matches β_1 until $b^{n_1} a b^{n_2} a$.

Keep getting $n_1, n_2, n_3, \dots, n_{m+1}$.
 $\alpha_m = b^{n_1} a b^{n_2} a \cdots b^{n_{m+1}} a b^\omega \in L$.

$\beta_m = \underbrace{\square_{\text{EG}} \square_{\text{EG}}}_{m+1 \text{ states}} \cdots \xrightarrow{\text{EG}}$

By PMP, two of these $m+1$ good states are equal. Loop between them to get a word which is accepted but has inf. many 'a's. \square

Cor. DBA \subsetneq NBA in terms of expressiveness.

Defn A language $L \subseteq \Sigma^\omega$ is said to be ω -regular if there exists a (possibly non-deterministic) Büchi automaton A such that $L(A) = L$.

CLOSURE PROPERTIES OF ω -REGULAR LANGUAGES

i) closure under union:

$$L_1 = L(A_1), \quad A_1 = (Q_1, q_0^1, \Sigma, \Delta_1, \delta_1),$$

$$L_2 = L(A_2), \quad A_2 = (Q_2, q_0^2, \Sigma, \Delta_2, \delta_2).$$

To-do: Construct a BA A s.t. $L(A) = L_1 \cup L_2$.

We do the usual product construction.

$$(Q_1 \times Q_2, (q_1^1, q_2^1), \Sigma, \Delta, \underbrace{G_1 \times G_2 \cup Q_1 \times b_2}_{\delta})$$

$$(q_1, q_2) \xrightarrow{a} (q_1', q_2')$$

If $q_1 \xrightarrow{a} q_1'$ and $q_2 \xrightarrow{a} q_2'$.

$$\alpha = c_0 a_1 a_2 \dots$$

$$s^1 = q_0' q_1' q_2' \dots \quad \text{a run of } A_1 \text{ on } \alpha$$

$$s^2 = q_0^2 q_1^2 q_2^2 \dots \quad \overbrace{\quad \quad \quad}^n \quad \overbrace{A_2 \quad \quad}^n$$

$$"s^1 s^2" = \begin{pmatrix} q_0' \\ q_0^2 \end{pmatrix} \begin{pmatrix} q_1' \\ q_1^2 \end{pmatrix} \begin{pmatrix} q_2' \\ q_2^2 \end{pmatrix} \dots \quad \text{a "product run" on } \alpha$$

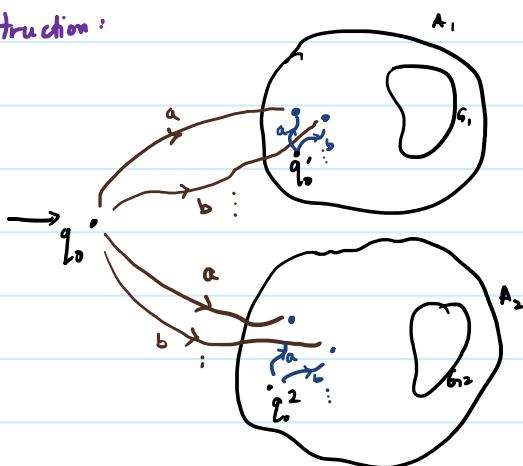
Here we assume that each $\alpha \in \Sigma^\omega$ has at least one

run on both A_i . (Can always ensure this by adding a dead state)

With the above assumption,

$G = (G_1 \times Q_2) \cup (Q_1 \times G_2)$ give
the language as $L_1 \cup L_2$.

A simpler construction:



Lecture 3 (06-08-2021)

06 August 2021 09:39

Closure under intersection.

Do the same product construction as earlier and put

$$G = G_1 \times G_2.$$

$$A = A_1 \times A_2.$$

Is: $L(A) = L(A_1) \cap L(A_2).$

(\Leftarrow) If $p = p_1 \times p_2$ is an accepting run, so p_1 and p_2 both are.

(\Rightarrow) Let $\alpha \in L(A_1) \cap L(A_2).$

Then there are accepting runs p_i on A_i .

$$\text{But } p = p_1 \times p_2.$$

But then it is not necessary that p is accepting.

For example, p_1 has good states at even positions and p_2 at odd.

As a concrete example of above:



$$\text{Then } (ab)^\omega \in (L(A_1) \cap L(A_2)) \setminus L(A_1 \times A_2).$$

Doesn't work! Slightly modified.

$$Q = Q_1 \times Q_2 \times \{1, 2\}, \quad \xrightarrow{\text{indicates}} \text{the component being "searched" for a good state}$$

$$q_0 = (q_0^1, q_0^2, 1)$$

$$\Delta = (q_1, q_2, 1) \xrightarrow{a} (q_1', q_2', 1) \text{ if } \begin{cases} q_1 \xrightarrow{a} q_1' & \text{if } q_1 \in G_1 \\ q_2 \xrightarrow{a} q_2' & \text{if } q_2 \in G_2 \end{cases}$$

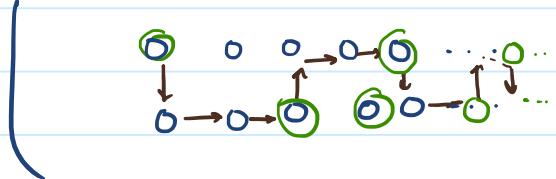
$$q_1 \notin G_1,$$

$$(q_1, q_2, 1) \xrightarrow{a} (q'_1, q'_2, 2) \quad \begin{array}{l} q_1 \xrightarrow{a} q'_1 \\ q_2 \xrightarrow{a} q'_2 \\ q_1 \in G_1 \end{array}$$

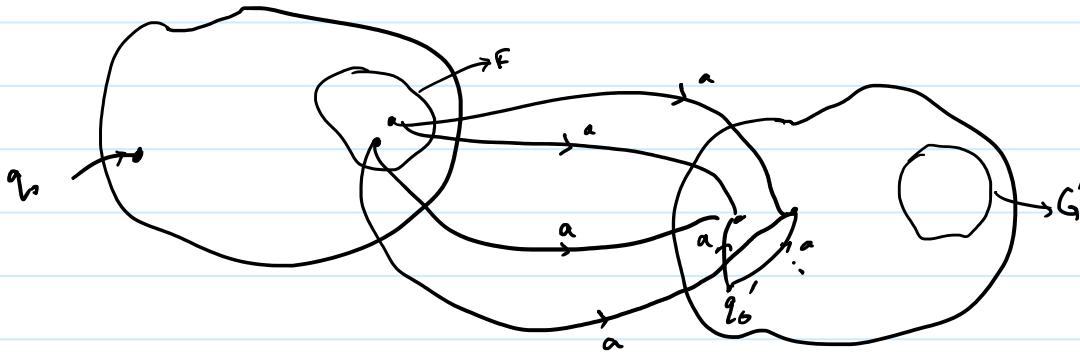
similarly for $(_, _, 2) \rightarrow (_, _, 2)$
 $(_, _, 2) \rightarrow (_, _, 1)$.

$$G = G_1 \times Q_2 \times \{1\}.$$

$$L(A) = L(A_1) \cap L(A_2).$$



Closure : $U \subseteq \Sigma^*$ regular $A = (Q_0, q_0, \Sigma, \Delta, F)$, $L(A) = U$
 $L \subseteq \Sigma^\omega$ ω -regular $B = (Q'_0, q'_0, \Sigma', \Delta', G)$, $L(B) = L$



Keep them disjoint and all possible transitions of the form:

$$q_f \xrightarrow{a} q'_f \quad \text{where } q_f \in F \text{ and } q'_f \xrightarrow{a} q'_f \text{ in } \Delta'$$

Keep G as G' .

Given $U \subseteq \Sigma^*$, define

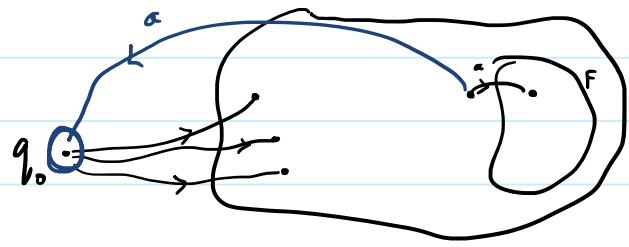
$U^\omega = \{ \alpha \in \Sigma^\omega : \alpha \text{ has a factorisation of the form}$

$$\alpha = \alpha_0 \alpha_1 \alpha_2 \alpha_3 \dots \text{ for } \alpha_i \in U.$$

Closure If $U \subseteq \Sigma^*$ is regular, then U^ω is ω -regular.

Let $A = (Q, q_0, \Sigma, \Delta, F)$ recognise U .

Assume that there are no incoming transitions to q_0 .
and that $q_0 \notin F$.
(Why can we do this?)
(Also note $U^\omega = (U \setminus \{\epsilon\})^\omega$)



(Also note $U^\omega = (U \setminus \{\epsilon\})^\omega$)

Add all possible transitions of the form:

$$q \xrightarrow{a} q_0 \quad \text{if} \quad \exists q_f \in F \text{ s.t. } q \xrightarrow{*} q_f.$$

Put $b_1 = \{q_0\}$.

Lecture 4 (11-08-2021)

11 August 2021 09:33

To Do: Closure under complementation.

Prop: Let L be ω -regular. Then, L can be expressed as

$$L = \bigcup_{i=1}^n U_i \cdot V_i^\omega,$$

where $U_i, V_i \subseteq \Sigma^*$ are regular languages for $i = 1, \dots, n$.

(By our earlier results, it is clear that any such L is indeed ω -regular.)

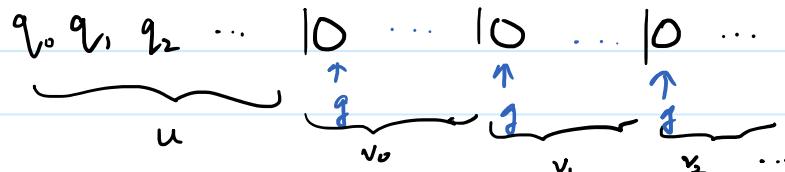
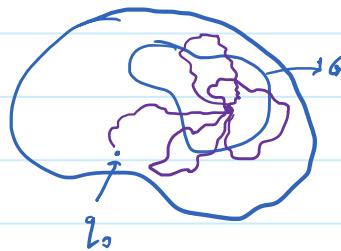
Proof: Let A be a BA s.t. $L(A) = L$.

$$(Q, \Sigma, q_0, \Delta, G)$$

Given an accepted word $w = a_0 a_1 a_2 \dots$

with an accepting run $\rho = q_0 q_1 q_2 \dots$,

$\exists g \in G$ which occurs i.o.



For $g \in G$, define

$$\begin{cases} U_g := \{w \in \Sigma^* : \exists \text{ a run } q_0 \xrightarrow{\omega} g\} \\ V_g := \{w \in \Sigma^* : \exists \text{ a run } g \xrightarrow{\omega} g\}. \end{cases}$$

regular since $A_g := (Q, \Sigma, q_0, \Delta, \{g\})$ and

$B_g := ((Q, \Sigma, \{g\}), \Delta, \{g\})$ accept them

Now, by our earlier observation, it is easy to argue that

$$L = \bigcup_{g \in G} U_g \cdot V_g^\omega.$$

Obs. The following problem is decidable: (Non emptiness problem)

- Input :— $A \rightarrow a \text{ BA}$
- Output :— YES if $L(A) \neq \emptyset$,
NO if $L(A) = \emptyset$.

$(L(A) \neq \emptyset \Leftrightarrow \exists g \in G \text{ st. } \exists q_0 \xrightarrow{\omega} g \text{ and } \exists g \xrightarrow{\omega} g)$

reachable from initial state
 both
 check if part of cycle
 efficient ✓

Obs. If $L(A) \neq \emptyset$, then there exist finite words u and v s.t. $|u|, |v| \leq |Q|$ and $u \cdot v^\omega \in L(A)$.

↑
 ultimately periodic

Let $A = (Q, \Sigma, q_0, \Delta, G)$ be a BA accepting L .

Goal: To show that $\bar{L} = \Sigma^\omega \setminus L$ is also ω -regular.

For $u, v \in \Sigma^*$, define

$$u \sim_p v \Leftrightarrow \forall q, q' \in Q, \quad q \xrightarrow{\omega} q' \text{ iff } q \xrightarrow{v} q' \text{ and} \\ q \xrightarrow{\omega_G} q' \text{ iff } q \xrightarrow{\omega} q'.$$

Notation: $s \xrightarrow{\pi_G} s'$ means
 $\exists \text{ a run on } \pi \text{ from } s \text{ to } s'$
 with an intermediate visit to G .

Observations:

(i) \sim_p is an equivalence relation on Σ^*

(i) \sim_A is an equivalence relation on Σ^* .

(ii) \sim_A is of finite index, i.e., it has finitely many equivalence classes.

Proof. Fix $q, q' \in Q$.

$$U_{q,q'} = \{w \in \Sigma^*: q \xrightarrow{w} q'\}$$

$$V_{q,q'} = \{w \in \Sigma^*: q \xrightarrow{w} q'\}$$

2^{n^2} such sets. ($n := |Q|$)

For each $u, v \in \Sigma^*$, we can ask 2^{n^2} questions about set membership. $u \sim_A v \Leftrightarrow$ they have same answers.

Thus, there are $\leq 2^{n^2}$ classes.

$$[u]_{\sim_A} = \left(\bigcap_{\substack{q, q' \in Q \\ u \in U_{q,q'}}} U_{q,q'} \right) \cap \left(\bigcap_{\substack{q, q' \in Q \\ u \in V_{q,q'}}} V_{q,q'} \right) \cap \left(\bigcap_{\substack{q, q' \in Q \\ u \notin U_{q,q'}}} \bar{U}_{q,q'} \right) \cap \left(\bigcap_{\substack{q, q' \in Q \\ u \notin V_{q,q'}}} \bar{V}_{q,q'} \right).$$

The above discussion also shows that each equivalence class is a regular language.

($U_{q,q'}$ is clearly regular. Some argument shows the same for $V_{q,q'}$.)

Let U_1, \dots, U_m be the equivalence classes of \sim_A .

Lemma: Suppose $L \cap (U_i \cdot U_j^\omega) \neq \emptyset$ for some i, j , then $U_i \cdot U_j^\omega \subseteq L$.

Proof. Let $\alpha \in L \cap (U_i \cdot U_j^\omega)$.

$$\alpha = \alpha_0 \alpha_1 \alpha_2 \alpha_3 \dots \in L.$$

Let $\rho = q_0 q_1 q_2 q_3 \dots$ be an accepting run of A

Let $\beta = q_0 q_1 q_2 q_3 \dots$ be an accepting run of A
on α .

We can also write $\alpha = u \cdot v_0 \cdot v_1 \cdot v_2 \dots$ s.t. $u \in U_i$ and
 $v_0, v_1, v_2, \dots \in U_j$.

$$\beta_\alpha = \underbrace{q_0}_{u}, \underbrace{q'_1}_{v_0}, \underbrace{q'_2}_{v_1}, \underbrace{q'_3}_{v_2} \dots$$

Now, let $\beta \in U_i \cup U_j^\omega$. Then, $\beta = u' v'_0 v'_1 \dots$

Then, we have a run

$$\beta_B = q_0 q'_1 q'_2 q'_3 \dots \text{ by def" of } \beta_A.$$

Moreover if β_A saw a good state $q'_i \xrightarrow{\alpha} q'_{i+1}$,

so does β_B .

$\therefore \beta \in L(A)$. B

Lecture 5 (13-08-2021)

13 August 2021 09:36

Ramsey's Theorem

$$\mathbb{N} = \{0, 1, 2, \dots\}$$

$$E = \{(i, j) \in \mathbb{N} \times \mathbb{N} : i < j\}$$

} Complete graph on \mathbb{N}

\mathcal{C} - a finite set of colours.

$x: E \rightarrow \mathcal{C}$ is called an edge-colouring of the complete graph on \mathbb{N} .

Thm. Given an arbitrary x , \exists an infinite monochromatic clique in x .
That is,

$\exists S \subseteq \mathbb{N}, |S| = \infty, \exists c \in \mathcal{C}$ such that every edge within S is coloured ' c '.

$$(\forall i, j \in S : i < j \Rightarrow x((i, j)) = c)$$

Proof. Fix $x: E \rightarrow \mathcal{C}$.

$$x_0 := \mathbb{N}, m_0 := \min(x_0) (= 0).$$



$\exists c \in \mathcal{C}$ s.t. \exists infinitely many x s.t.
 $x((m_0, x)) = c$.

Let $x_1 := \{\text{neighbours of } m_0 \text{ in } x_0\} \subseteq x_0 \setminus \{m_0\}$.

Note: $x_1 \subseteq \mathbb{N}$ is infinite.

Let $m_1 := \min(x_1)$ and proceed similarly to pick
 c and $x_2 \subseteq x_1 \setminus \{m_1\}\dots$

In general, we have an infinite subset x_{k+1} and colour c_k
s.t. every element of x_{k+1} is connected to $\min(x_k)$ by c_k .

Define $x_\infty := \{m_0, m_1, m_2, \dots\}$. $(m_0 < m_1 < m_2 < \dots)$

Then, x_∞ is an infinite set s.t. $\forall i, j : x((m_i, m_j)) = c_i \quad \forall i < j$.

As usual, $\exists c \in \mathcal{C}$ which occurs infinitely many often.

Simply restrict graph to these vertices.



Continuing from last lecture: U_1, \dots, U_m are equiv. classes of \sim_A .

We know: U_i are regular.

Claim. $\Sigma^\omega = \bigcup_{i,j} U_i \cdot U_j^\omega$.

Proof. Only (\subseteq) is to be shown.

Let $\alpha \in \Sigma^\omega$ be arbitrary.

TS: $\stackrel{\exists i, j}{\alpha} = u_0 v_0 v_1 v_2 \dots$ for $u_0 \in U_i$ and $v_k \in U_j \forall k$.

Write $\alpha = a_0 a_1 a_2 a_3 \dots \in \Sigma^\omega$ for $a_i \in \Sigma$.

Define the coloring χ_α on (\mathbb{N}, \in) as:

$$\mathcal{C} = \{U_1, \dots, U_m\}$$

$$\chi_\alpha(i, j) = [a_i a_{i+1} \dots a_{j-1}]_{\sim_A}.$$

\hookrightarrow equiv class of $a_i a_{i+1} \dots a_{j-1}$

By Ramsey's theorem, $\exists U_j$ with a clique, i.e., $\exists m_1 < m_2 < m_3 < \dots$
s.t. $\chi_\alpha((n_k, n_{k+1})) = U_j \quad \forall j$.

Defining

$$u_0 = a_0 \dots a_{m_1-1}, \quad v_0 = a_{m_1} \dots a_{m_2-1}, \\ v_1 = a_{m_2} \dots a_{m_3-1}, \dots$$

does the job. \square

$$\Sigma^* = U_1 \sqcup U_2 \sqcup \dots \sqcup U_m,$$

$$\Sigma^\omega = \bigcup_{i,j} U_i \cdot U_j^\omega.$$

Note that U_i are regular. Moreover, we have

$$L \cap (U_i \cdot U_j^\omega) \neq \emptyset \Rightarrow U_i \cdot U_j^\omega \subseteq L.$$

Thus, $L = \bigcup_{\text{some } i, j} U_i \cdot U_j^\omega.$

Thus, $\sum^\omega \setminus L = \bigcup_{i, j : U_i \cdot U_j^\omega \not\subseteq L} U_i \cdot U_j^\omega.$

Thus, it is again ω -regular. \square

→ Effective construction of BA for \bar{L} .

- Construct automaton for U_i .
- Construct BA for $U_i \cdot U_j^\omega$.
- Take union of those not in L .

(Can effectively check if $L \cap (U_i \cdot U_j^\omega) = \emptyset$.)

Lecture 6 (18-01-2021)

18 August 2021 09:31

Büchi's Theorem:

Want to talk about properties of words (finite or infinite).

First-order Logic (over words)

Fix $\Sigma \rightarrow \text{alphabet}$.

First-order variables - $x, y, z, x_1, x_2, x_3, \dots$

Range over positions
in the word

Atomic-predicate - $a(x), b(x), \dots$

→ unary predicate

for $a, b, \dots \in \Sigma$

and x is a Fo variable.

$x < y$

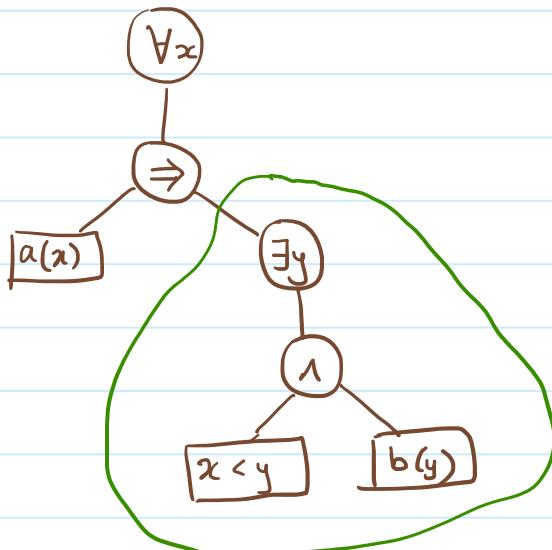
→ binary predicate

Syntax :

$\varphi \equiv a(x) \mid x < y \mid \neg \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi \mid \forall x. \varphi$

derived : $\varphi_1 \Rightarrow \varphi_2 \equiv \neg \varphi_1 \vee \varphi_2$

Example : $\varphi_1 \equiv \forall x. [a(x) \Rightarrow \exists y (x < y) \wedge b(y)]$



Semantics :

$\varphi(x_1, \dots, x_m) \quad - \quad \varphi \text{ is a formula with free variables}$

$\varphi(x_1, \dots, x_m) — \varphi$ is a formula with free variables

x_1, \dots, x_m

$$\varphi' = \exists y[(x < y) \wedge b(y)] \rightarrow x \text{ free, } y \text{ bound}$$

$\varphi(x_1, \dots, x_m), w \rightarrow \text{word}$

$$w, x_1 \leftarrow p_1, \dots, x_m \leftarrow p_m \models \varphi(x_1, \dots, x_m)$$

defined by structural induction

1. " $w, x_i \leftarrow p_i \models a(x_i)$ " iff the letter in w at position p_i is a

2. " $w, x_1 \leftarrow p_1, x_2 \leftarrow p_2 \models x_1 < x_2$ " $\Leftrightarrow p_1 < p_2$

:

Example. ① $\varphi_2 \equiv \exists x \forall y [(x < y) \Rightarrow \neg a(y)].$

If w is finite, then it will satisfy φ_2 .

But if w is infinite, then $w \models \varphi_2 \Leftrightarrow w$ has finitely many 'a's

② $\varphi_3 \equiv \forall x \exists y (x < y)$

if w is a (nonempty) finite word, then $w \not\models \varphi_3$.

OTOH, all infinite words satisfy this property

Büchi - Elgot Theorem \rightarrow a logical characterisation of (finite) regular languages

Büchi Theorem \rightarrow a logical characterisation of ω -regular languages

Monadic Second-Order Logic over words

Extends F0 - over words

position variables - $x, y, x_1, x_2, x_3, \dots$

sets-of-positions variables - $X, Y, X_1, X_2, X_3, \dots$

atomic-predicate - $a(x), x < y, X(x)$.

Syntax

$\varphi \equiv \text{atomic-predicates} \mid \neg \varphi \mid (\varphi \vee \varphi) \mid \varphi \wedge \varphi \mid \exists x \cdot \varphi \mid \forall x \cdot \varphi \mid \exists X \cdot \varphi \mid \forall X \cdot \varphi$

$S(x, y) \equiv$ position y is successor of position x
 $\equiv (x < y) \wedge \neg (\exists z \cdot (z < x) \wedge (z < y))$.

$\text{first}(x) \equiv x$ is the first position
 $\equiv \forall y \cdot (x = y \vee x < y)$.

$\text{last}(x) \equiv \dots$

Remark. In F_0 , the ' $<$ ' predicate cannot be expressed using ' S ' predicate.

$x < y \Leftrightarrow$ $x \neq y$ and
 $x < y \Rightarrow$ every successor-closed set of positions which
contains x , also contains y
 \hookrightarrow can define in $M\sigma$

Thus, we can write ' $<$ ' in terms of ' S ' in $M\sigma$.

Defn. Let φ be a $M\sigma$ sentence.

$$L_\varphi = \{\alpha \in \Sigma^\omega : \alpha \models \varphi\}.$$

$L \subseteq \Sigma^\omega$ is called $M\sigma$ -definable if $\exists M\sigma \varphi$ s.t. $L = L_\varphi$.

Theorem (Büchi's Theorem)

Let $L \subseteq \Sigma^\omega$.

L is $M\sigma$ -definable $\Leftrightarrow L$ is ω -regular.

Lecture 7 (20-08-2021)

20 August 2021 09:37

Thm. (Buchi) Let $L \subseteq \Sigma^\omega$.

L is ω -regular $\Leftrightarrow L$ is MSO-definable.

Proof. (\Rightarrow) Suppose L is ω -regular, say $L = L(A)$, where $A = (Q, q_0, \Sigma, \Delta \subseteq Q \times \Sigma \times Q, G)$ is a BA.

Goal: Construct MSO sentence φ_A s.t.

$$\forall \alpha \in \Sigma^\omega : \alpha \models \varphi_A \Leftrightarrow A \text{ accepts } \alpha.$$

$$\alpha = a_0 a_1 a_2 a_3 a_4 \dots$$

Suppose A accepts α via an accepting run ρ .

$$\rho = q_0 q_1 q_2 q_3 q_4 \dots$$

$\forall q \in Q, X_q \equiv$ The set of positions in α when
the run ρ is in the state ' q '
 $= \{i \in \mathbb{N} : q_i \in q\}$. $(0 \in \mathbb{N})$

Note that $\{X_q\}_{q \in Q}$ is a partition of \mathbb{N} .
(Allowing \emptyset in partition.)

- 1) $0 \in X_{q_0}$
- 2) for any two consecutive positions x and y ,
if $x \in X_q$, $y \in X_{q'}$, then $(q, a, q') \in \Delta$,
where a is the letter at position x .
- 3) for any position x , there is a position y
to the right of x such that $y \in X_q$ for
some $q \in G$.

Conversely, given a partition with above 3 properties, we
can build an accepting run.

For convenience, write $Q = \{0, 1, \dots, m\}$.

$$\varphi_A = \exists x_0 \exists x_1 \dots \exists x_m \cdot \text{partition}(x_0, \dots, x_m) \wedge$$

$$[\forall x \cdot \text{first}(x) \Rightarrow x_0(x)] \wedge$$

$$[\forall x \forall y \quad S(x, y) \Rightarrow \left(\bigvee_{(i, j) \in A} x_i(x) \wedge x_j(y) \wedge c(i) \right)] \wedge$$

$$[\forall x \exists y \quad (x < y) \wedge \left(\bigvee_{i \in A} x_i(y) \right)]$$

$$\text{partition}(x_0, x_1, \dots, x_m)$$

$$\equiv \forall x \left(\bigvee_{i \in A} x_i(x) \wedge \bigwedge_{i \neq j} \neg(x_i(x) \wedge x_j(x)) \right)$$

$$\text{length}(\varphi_A) = O(|A|).$$

(\Leftarrow) Given: MSO sentence φ .

Goal: Construct BA A s.t.

$$L(A) = \{\alpha \in \Sigma^\omega \mid \alpha \models \varphi\}.$$

As in the finite case, we use MSO₀-logic

\hookrightarrow substitute position variables by singleton set vars.
 \hookrightarrow more atomic predicates: $\text{Sing}(x)$, $a(x)$, $S(x, y)$, $x \leq y$
 ↓
 singleton
 ↓
 x, y are sing
 and the single
 positions are related
 by S
 these can be defined
 in MSO.

\hookrightarrow The converse is true too. Thus, we use them interchangeably.
 \hookrightarrow That is, they have some expressive power.

Lecture 8 (25-08-2021)

25 August 2021 09:39

Goal: Given a MSO₀ formula, construct a BA A s.t. $L(A) = L(\varphi)$.

The construction of A proceeds by structural induction φ .

In fact: let $\varphi(x_1, \dots, x_n)$ be an MSO₀-formula with free (set-) variables x_1, \dots, x_n

$$\alpha, x_1 \leftarrow p_1, \dots, x_n \leftarrow p_n \models ? \varphi$$

$$\begin{array}{ll} \alpha = a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ \dots & \\ \text{P}_1 = \{0, 1, 0, \dots\} & \\ \text{P}_2 = \{1, 0, 0, 1, 0, \dots\} & \\ \vdots & \\ \text{P}_n = \{0, 1, 0, 1, 0, \dots\} & \end{array} \quad \left. \begin{array}{l} \text{characteristic} \\ \text{vectors} \end{array} \right\}$$

$$\begin{pmatrix} a_0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} a_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} a_2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} a_3 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} a_4 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \dots$$

The model of φ (an inf. word Σ + n sets)
can be seen as an inf. word over $\Sigma \times \{0, 1\}^n$.

$$\text{Free } (\varphi) = \{x_1, \dots, x_n\}.$$

$$L(\varphi) = \{ \alpha' \in (\Sigma \times \{0, 1\}^n)^\omega : \alpha' \models \varphi \}.$$

Claim: $L(\varphi)$ is ω -regular over $\Sigma_n = \Sigma \times \{0, 1\}^n$.

Root: $\varphi \leadsto A_\varphi$ by structural induction.

base. φ - atomic predicate

$$\varphi = \text{Sing}(x_i)$$

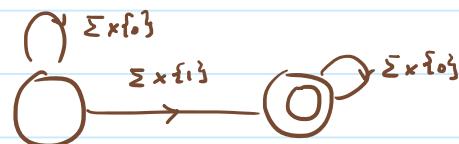
$$\alpha' = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \dots$$

$$\begin{array}{l} a_i \in \Sigma \\ b_i \in \{0, 1\} \end{array}$$

$$\varphi = \text{Sing}(x_1)$$

$$\alpha' = \left(\begin{smallmatrix} a_0 \\ b_0 \end{smallmatrix} \right) \left(\begin{smallmatrix} a_1 \\ b_1 \end{smallmatrix} \right) \left(\begin{smallmatrix} a_2 \\ b_2 \end{smallmatrix} \right) \dots$$

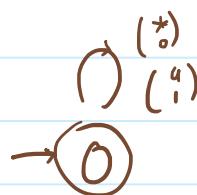
$$\begin{array}{l} a_i \in \Sigma \\ b_i \in \{0,1\} \end{array}$$



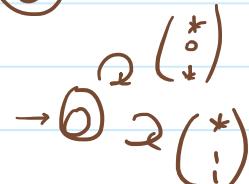
$$\varphi = S(x_1, x_2) \quad ; \quad \Sigma_2 = \Sigma_x \cup \{0,1\} \cup \{f_0,1\}$$



$$\varphi = a(x_1)$$



$$\varphi = x_1 \leq x_2$$



Inductive step.

$$\varphi = \varphi_1 \vee \varphi_2.$$

$$\text{Free}(\varphi) \subseteq \{x_1, \dots, x_n\}.$$

wlog, we may assume $\text{Free}(\varphi_1) = \{x_1, \dots, x_n\}$.
 $\text{Free}(\varphi_2)$

By induction, we have appropriate automata A_{φ_i} for φ_i .

But the alphabet for both is same. Can take union of $S A_{\varphi_i}$.

$$\varphi = \varphi_1 \wedge \varphi_2, \quad \varphi = \neg \varphi_1, \quad \text{similarly done.}$$

$$\varphi = \exists x_n \varphi'(x_1, \dots, x_n)$$

$$\text{Free}(\varphi) = \{x_1, \dots, x_{n-1}\}$$

Note : $\alpha' \models \varphi \Leftrightarrow \exists \text{ a set } P_n \text{ st. }$

$$\alpha', x_n \leftarrow P_n \models \varphi'$$

Consider the projection map

$$\pi : \Sigma \times \{0, 1\}^n \rightarrow \Sigma \times \{0, 1\}^{n-1},$$

$$(a, b_1, \dots, b_n) \mapsto (a, b_1, \dots, b_{n-1}).$$

This induces a map $\pi : (\Sigma^n)^\omega \rightarrow (\Sigma^{n-1})^\omega$.

$$\alpha' \in (\Sigma^{n-1})^\omega, \quad \alpha' \models \varphi \iff \exists \alpha'' \in (\Sigma^n)^\omega \text{ s.t.}$$

$$\pi(\alpha'') = \alpha' \text{ and } \alpha'' \models \varphi'.$$

The question is reduced to asking if projection of an ω -regular language is ω -regular.

But this is simple to see. Take an automaton for φ' and erase the last coordinate on all transitions.

$$\bullet \varphi = \forall X_n. \varphi'.$$

$$\text{Some } a_0 \exists X_n \rightarrow \varphi'.$$

B

Thus, we are done.

Thm.

(Büchi's Theorem) let $L \subseteq \Sigma^\omega$. Then,

$$L \text{ is } \omega\text{-regular} \iff L \text{ is MSO-definable.}$$

Moreover, the translations are effective.

The above theorem was proven a few years after the Büchi - Elgot theorem (the analogous theorem about (finite) regular languages).

It is easy to see how MSO-definability translates to ω -words but was not so clear how to extend regularity.

Thus, $\text{MSO}(\Sigma)$ is decidable.

Given an MSO sentence φ , we can check if there

exists an inf. word $\alpha \in \Sigma^\omega$ s.t. $\alpha \models \varphi$.

$\left[\varphi \rightsquigarrow A_\varphi$ is effective and we can check $L(A_\varphi) \neq \emptyset \right]$

In fact, if $L(A_\varphi) \neq \emptyset$, then $\exists u, v$ s.t. $uv^\omega \in L(A_\varphi)$

and we can produce the above u, v .

Note: $\varphi \rightsquigarrow A_\varphi$ is non-elementary.

We cannot bound $|A_\varphi|$ in terms of any

(fixed) k -ary exponential of $|\varphi|$.

$(n = |\varphi|, 2^{P(n)}, 2^{2^{P(n)}}, \dots \leftarrow \text{elementary})$

singly exp

doubly exp

$P \leftarrow \text{polynomial}$

The tower (we get) will have length in terms of n .

↙ can we do better for satisfiability?

FACT. There is a non-elementary type lower bound for MSO-satisfiability problem.

Note: $\varphi \rightsquigarrow A_\varphi \rightsquigarrow \varphi_{A_\varphi}$

↳ This has a nice form

$\exists x_1 \dots \exists x_n$ ("first-order type").

Lecture 9 (27-08-2021)

27 August 2021 09:34

"First-order theory" of arithmetic

$$(\mathbb{N}, +, \cdot, 0, 1)$$

- $\text{add}(x, y, z) \rightarrow \text{asserts } x + y = z$
- $\text{mult}(x, y, z) \rightarrow \text{asserts } xy = z$

Usual Fo syntax.

- $\text{zero}(x) \equiv \text{add}(x, x, x)$
- $x < y \equiv \exists z \text{ add}(x, z, y)$
- $S(x, y) \equiv x < y \wedge \exists z (x < z \wedge z < y)$
- $\text{one}(x) \equiv \exists y (\text{zero}(y) \wedge S(y, x))$
- $\text{prime}(x) \equiv \neg \text{one}(x) \wedge \forall y \forall z (\text{mult}(y, z, x) \Rightarrow \text{one}(y) \vee \text{one}(z))$
- $\text{even}(x) \equiv \exists y \text{ add}(y, y, x)$.

Goldbach's conjecture:

$$\psi \equiv \forall z \text{ even}(z) \Rightarrow \exists y \exists x \text{ prime}(x) \wedge \text{prime}(y) \wedge \text{add}(x, y, z).$$

Given a sentence ψ , we would like to know if ψ is true.

Hilbert's belief: Perhaps, we can mechanically figure out the truth/falsity of ψ .

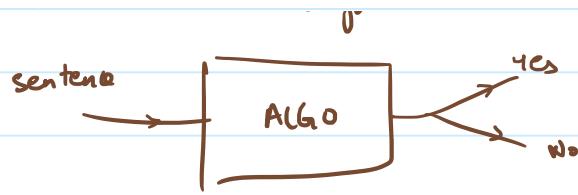
? what does this mean?

Church/ Gödel/ Turing : "Computability" ← defined

Moreover, it was shown that ·

The first-order theory of arithmetic is undecidable/
non-computable

That is, there is no algorithm s.t.



Now, let us look at $(\mathbb{N}, +) \rightarrow$ Presburger arithmetic.

First order and only add (x, y, z).

This IS decidable!

$(\mathbb{N}, +, <) \rightarrow$ first-order theory

Büchi showed that Presburger arithmetic is decidable using automata theory

$\varphi(x_1, \dots, x_n) \rightarrow$ first-order formula

encode x_1, \dots, x_n in reverse binary order
finite words over $\{0, 1\}^n$.

$\rightarrow \underline{\text{S1S}}$: $(\mathbb{N}, <) \text{ or } (\mathbb{N}, S)$

second order theory of 1 successor

Both same if monadic second order

(In fact, MSO + add gives mult, which we know is undecidable)

1960s:

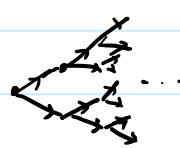
WS1S : Subsets are only allowed to be finite subsets
weak

Büchi showed that S1S is decidable

S2S is decidable

(Rabin's theorem)

two successors



MONA : Logic \rightarrow Automata

New modes of acceptance for automata on infinite words

$A : (Q, \Sigma, q_0, \Delta \subseteq Q \times \Sigma \times Q, A \alpha)$

Büchi : $G \subseteq Q$, ρ is accepting if $\text{Inf}(\rho) \cap G \neq \emptyset$.

Muller : $\mathcal{F} = \{F_1, \dots, F_k\}$ is a collection of subsets of Q .

ρ is accepting if $\text{Inf}(\rho) = F_i$ for some i .

\downarrow
"states at ∞ "

Rabin : $\Omega = \{(E_1, F_1), \dots, (E_k, F_k)\}$, each $E_i, F_i \subseteq Q$.

ρ is accepting if

$\exists i : \text{Inf}(\rho) \cap E_i = \emptyset$ and $\text{Inf}(\rho) \cap F_i \neq \emptyset$.

$$\bigvee_{i=1}^k [(\text{Inf}(\rho) \cap E_i = \emptyset) \wedge (\text{Inf}(\rho) \cap F_i \neq \emptyset)].$$

Streett : Dual to Rabin.

$\Omega = \{(E_1, F_1), \dots, (E_k, F_k)\}$

ρ is accepting if

$$\bigwedge_{i=1}^k [(\text{Inf}(\rho) \cap E_i \neq \emptyset) \vee (\text{Inf}(\rho) \cap F_i = \emptyset)]$$

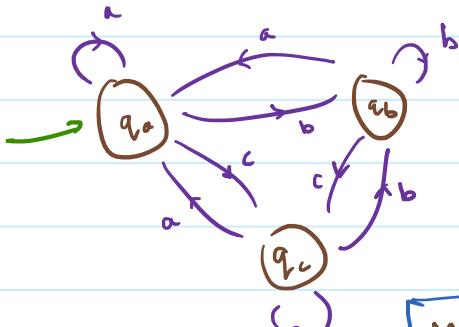
$\underbrace{\quad}_{\text{equivalently}}$

$$= \bigwedge_{i=1}^k ([\text{Inf}(\rho) \cap F_i \neq \emptyset] \Rightarrow [\text{Inf}(\rho) \cap E_i \neq \emptyset]).$$

↳ If F_i is visited infinitely often, then
so is E_i .

$$\Sigma = \{a, b, c\}$$

$L = \{\alpha \in \Sigma^\omega \mid \text{if 'a' occurs inf. often in } \alpha, \text{ then 'b' does inf. often}\}$



Street-condition:

$$\Omega = \{(\{q_a\}, \{q_a\})\}.$$

Rabin-condition

Note:

$$L = \{\alpha \mid b \text{ occurs inf. often}\} \cup \{\alpha \mid \text{both } a \text{ & } b \text{ fin. often}\}$$

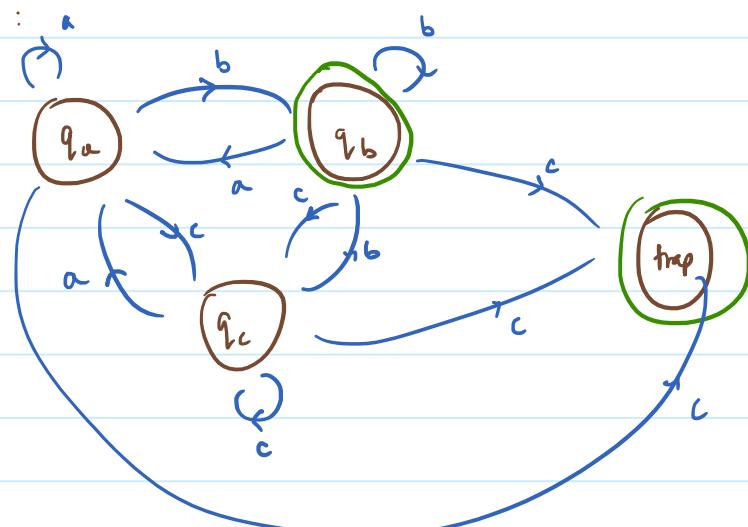
$$\Omega = \{(\emptyset, \{q_b\}), (\{q_a, q_b\}, \emptyset)\}$$

Muller-condition

$$\begin{aligned} F &= \{\{q_a, q_b\}, \{q_a, q_b, q_c\}, \\ &\quad \{q_b\}, \{q_b\}, \{q_b, q_c\}\} \\ &= \{X \subseteq Q : q_{a,b} \in X \Rightarrow q_b \in X\} \setminus \{\emptyset\}. \end{aligned}$$

Note that putting \emptyset in F makes no difference since $\inf(\emptyset) \neq \emptyset$ vs.

Büchi-condition:



Thm (McNaughton)

DMA = DRA = DSA = NBA = ω -regular.

D = deterministic

N = non-D

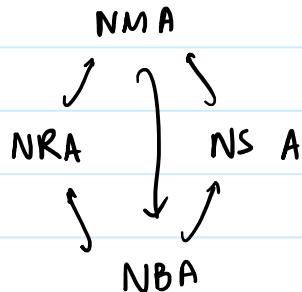
M = Muller, R = Rabin, S = Streetter, B = Büchi

A = Automaton

Lecture 10 (03-09-2021)

01 September 2021 09:58

Thm.



Thus, all are equivalent
in terms of expressive power.

($X \hookrightarrow Y$: any lang. acc. by X
can also be acc. by Y)

Proof.

• $NBA \hookrightarrow NRA$

Let $A = (Q, \Sigma, q_0, \Delta, \delta)$ $\hookleftarrow NBA$.

For Robin: Put $S_2 = \{(q_0, \delta)\}$.

• $NBA \hookrightarrow NSA$.

For A as above, put $S_2 = \{(q_0, Q)\}$.

For Muller: $F = \{x \in Q : x \cap G \neq \emptyset\}$.

• $NRA \hookrightarrow NMA$.

Let $A = (Q, \Sigma, q_0, \Delta, \delta)$ be a NRA.

$\delta = \{(E_1, F_1), \dots, (E_k, F_k)\}$.

$F = \{x \in Q : \exists i \text{ s.t. } x \cap E_i = \emptyset \text{ and } x \cap F_i = \emptyset\}$.

• $NSA \hookrightarrow NMA$: similar.

• $NMA \hookrightarrow NBA$.

Let $A = (Q, \Sigma, q_0, \Delta, F)$ be a NMA.

$F = \{x_1, \dots, x_m\}$.

For $i \in [m]$, define $A_i := (Q, \Sigma, q_0, \Delta, \{x_i\})$.

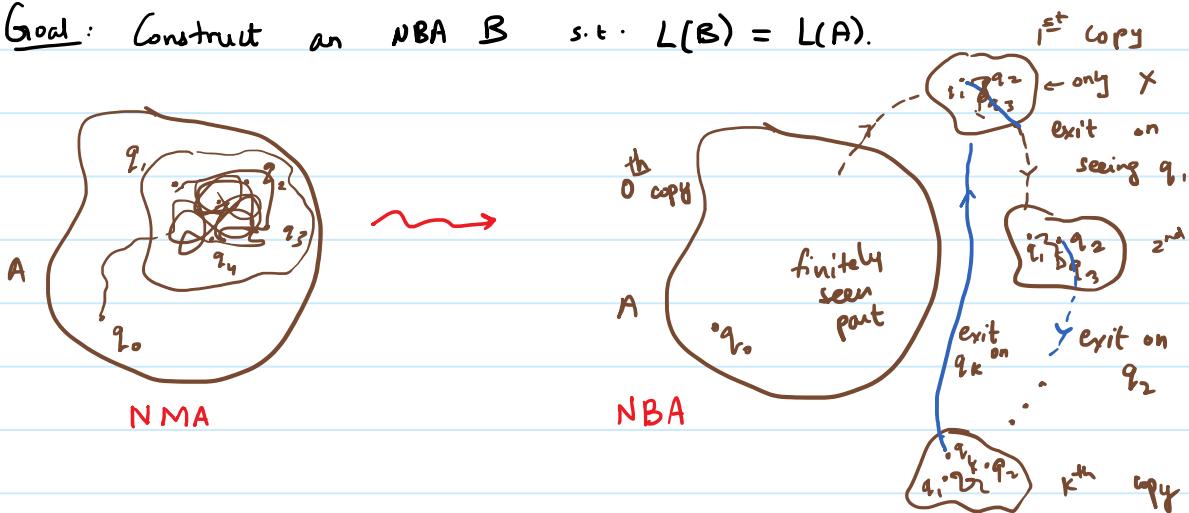
Note: $L(A) = \bigcup_{i=1}^k L(A_i)$.

Since ω -reg. languages are closed under union, suffice to

translate each A_i to an NBA. Equivalently, we may assume $m = 1$.

$$A = (Q, \Sigma, q_0, \Delta, \{x\}), \text{ and write } x = \{q_1, \dots, q_k\}.$$

Goal: Construct an NBA B s.t. $L(B) = L(A)$.



$$B = (Q', (q_0, 0), \Sigma, \Delta', \{(q_1, 1)\})$$

$$Q' = (Q \times \{0\}) \cup (\times \times \{1, 2, \dots, k\}).$$

$$\begin{aligned} \Delta' : & (q_1, 0) \xrightarrow{a} (q'_1, 0) \quad \text{if } q \xrightarrow{a} q'_1 \\ & (q_1, 0) \xrightarrow{a} (q'_1, 1) \quad \text{if } q \xrightarrow{a} q'_1 \text{ and } q'_1 \in x \\ & q, q' \in x \rightarrow (q, i) \xrightarrow{a} (q', i) \quad \text{if } q \neq q_i \text{ and } q \xrightarrow{a} q' \\ & (q, i) \xrightarrow{a} (q', i+1) \quad \text{if } q = q_i \text{ and } q \xrightarrow{a} q' \end{aligned}$$

$\sum_{k=1}^k$

↑
the arrows here
indicate those in
original Δ

Thus, all these models have same expressive power.

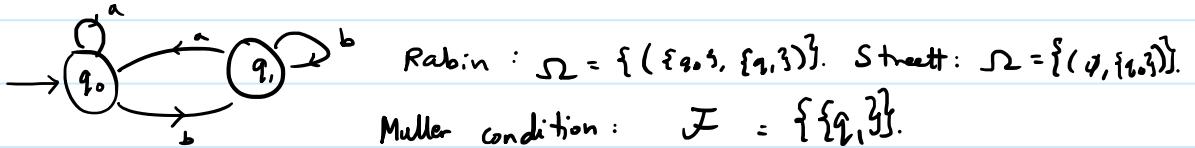
$$NRA = NFA = NSA = NBA$$

But

DBA

We had seen $L = \{\alpha \mid \alpha \text{ has finitely many 'a's}\} \in NBA \setminus DBA$.
(on $\Sigma = \{a, b\}$)

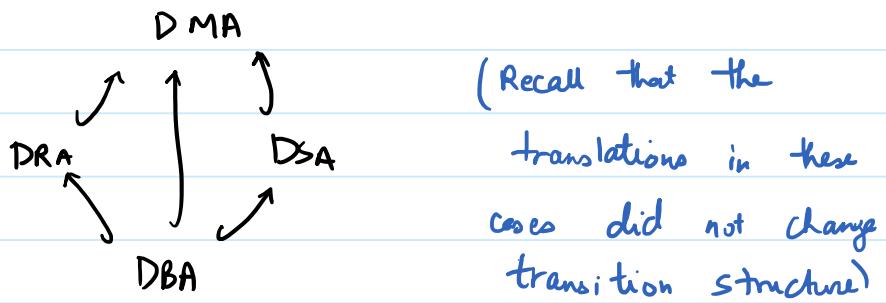
We had seen $L = \{\alpha \mid \alpha \text{ has finitely many 'a's}\} \in \text{NSA} \setminus \text{DBA}$.
 (on $\Sigma = \{a, b\}$)



Rabin: $\mathcal{R} = \{\{\{q_0\}, \{q_1\}\}\}$. Streett: $\mathcal{S} = \{\langle q, \{q_0\} \rangle\}$.
 Muller condition: $\mathcal{F} = \{\{q_1\}\}$.

Then, L is accepted by this DFA.

In fact, from the proof of the last theorem, we get



The L above shows that $\text{DMA} \not\hookrightarrow \text{DBA}$.

The following is true:

$$\text{NMA} \equiv \text{NRA} \equiv \text{NSA} \equiv \text{NBA}$$

$$\text{|||} \quad \text{|||} \quad \text{|||} \quad \text{JHR}$$

$$\text{DMA} \equiv \text{DRA} \equiv \text{DSA} \not\equiv \text{DBA}$$

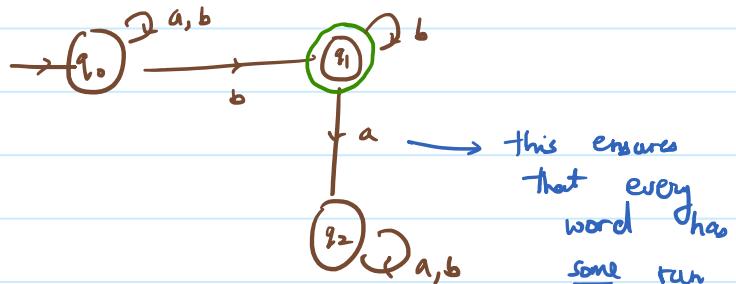
Lecture 11 (03-09-2021)

03 September 2021 09:28

Determinisation of Büchi automata:

Let $A = (Q, \Sigma, q_0, \Delta, G)$ be a NFA.

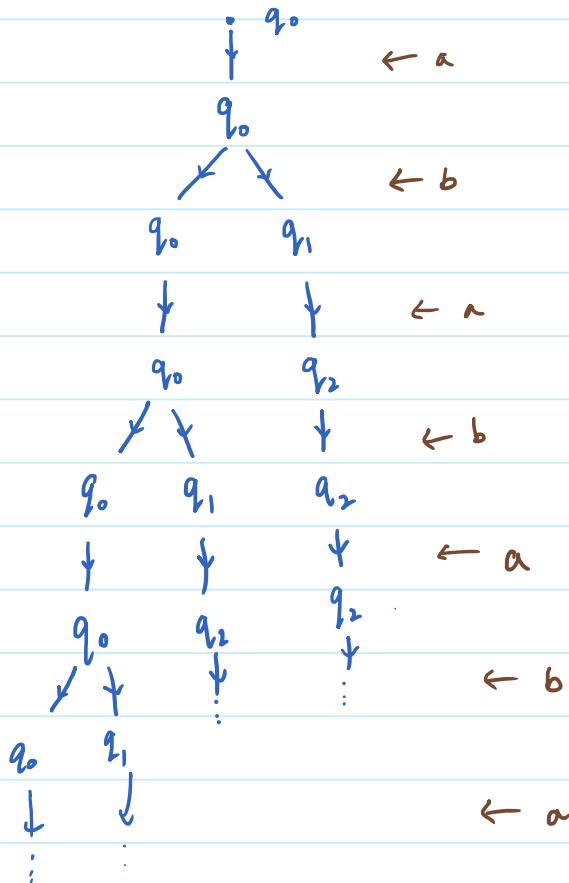
Finitely many 'a's :



Want : deterministic automaton accepting the above.

All runs of A on $\alpha \in \Sigma^{\omega}$ can be seen as a "run-tree" of A on α .

For example, take $\alpha = (ab)^\omega$.



A run of α on π corresponds to an inf

path in this run-tree.

Büchi acceptance $\equiv \exists$ an inf path where a good state appears infinitely often.

The subset / powerset construction:

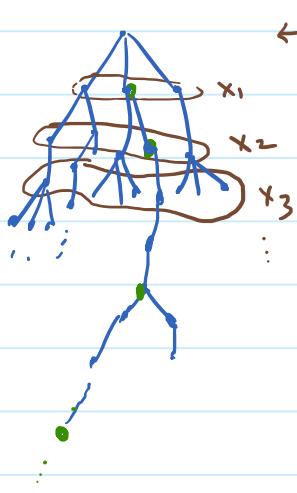
$$A_s = (2^Q, \Sigma, \{q_0\}, \delta_s, \text{acceptance}).$$

$\delta_s : 2^Q \times \Sigma \rightarrow 2^Q$ is defined as

$$\delta_s(x, a) := \{ q \in Q : \exists q' \in x, (q', a, q) \in \Delta \}.$$

(Look at x and take all states q' s.t. $q' \xrightarrow{a} q$ for some $q' \in x$)

Suppose A accepts the word $\alpha = a_0 a_1 a_2 a_3 \dots$



$\leftarrow x_0 = \{q_0\}$ The run of α on A_s looks like: $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} x_3 \xrightarrow{\dots}$

There are infinitely many indices i s.t. x_i contains a good state.

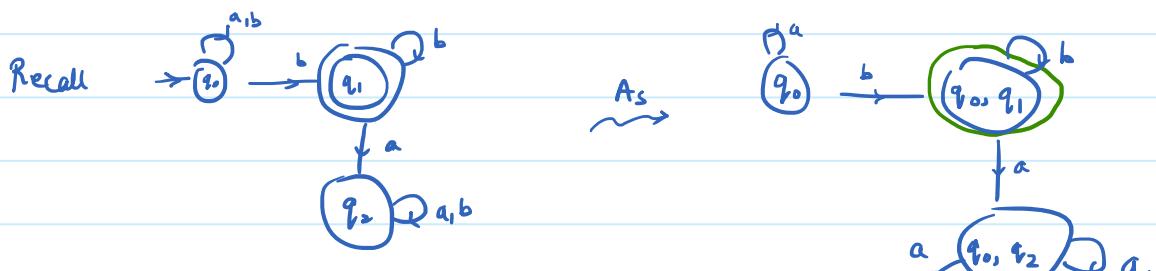
Define

$$G' := \{ x \subseteq Q : x \cap G \neq \emptyset \} \subseteq 2^Q.$$

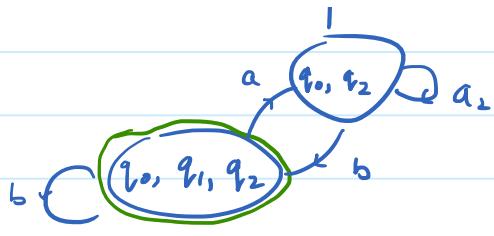
↳ good states

Observation: $L(A) \subseteq L(A_s, G')$.

But in general, $L(A) \subsetneq L(A_s, G')$ as we've already seen $DFA \not\equiv NFA$.



$(q_2) \xrightarrow{a,b}$



For $(ab)^\omega$, good states appear infinitely many often.

(Along the original tree, there were infinitely many levels with a good state but no single path with infinitely many good states.)

Key property: If $X \xrightarrow{u} Y$ in A_S , then $\forall q \in Y \exists q' \in X$ s.t. $q' \xrightarrow{u} q$.

A more refined "acceptance" condition on the subset automaton:

Defn $X \xrightarrow{u}_G Y$ if $\forall q \in Y \exists q' \in X$ s.t.

$q' \xrightarrow{u} q$. (There is a run of A on u from q' to q which visits a good state.)

A run of A_S on u

$x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} x_3 \rightarrow \dots$

is said to be strongly accepting if $\exists i_0 < i_1 < i_3 < \dots$
s.t.

$x_{i_0} \xrightarrow{u_1}_G x_{i_1} \xrightarrow{u_2}_G x_{i_2} \xrightarrow{u_3}_G \dots$

and $a_0 a_1 a_2 \dots = u_0 u_1 u_2 \dots$

Lemma: If A_S strongly accepts $\alpha \in \Sigma^\omega$, then A accepts α .

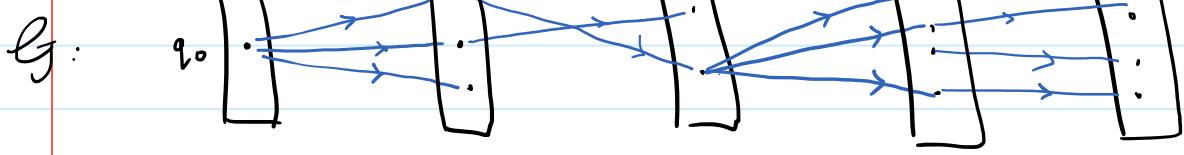
Proof:

$\alpha = u_0 u_1 u_2 u_3 \dots$

$v \xrightarrow{u_0} v \xrightarrow{u_1} v \xrightarrow{u_2} v \xrightarrow{u_3} v \xrightarrow{u_4} \dots$

$X_0 \xrightarrow{u_0} X_{i_0} \xrightarrow{u_1} \underset{\in}{\sim} X_{i_1} \xrightarrow{u_2} \underset{\in}{\sim} X_{i_2} \xrightarrow{u_3} \underset{\in}{\sim} X_{i_3} \xrightarrow{u_4} \underset{\in}{\sim} \dots$

 Write in terms of states



All the arrows here pass through a good state.
except at first level

To show: \exists an infinite path in the directed graph \mathcal{G} .

Obs. 1: Every vertex of \mathcal{G} is reachable from initial vertex.

Question. Let T be an infinite tree s.t. every vertex has finite deg.

Does T contain an infinite path?

That is, does \exists an inf seq v_0, v_1, v_2, \dots

s.t. v_i is a child of v_{i-1} ?

Ans. Yes. Keep picking a child s.t. the subtree below it

is infinite. (Such a child exists since tree is infinite and # children is finite)



König's Lemma (A finitely branching infinite tree must have an infinite path.)

Applying the lemma to our graph yields the result. \square

Q. How to implement the "stronger acceptance"?

Marked - subset automaton:

$X \rightarrow$ the set of reachable states

$Y \rightarrow$ the set of states which can be

reached via good states

$$A_m = (2^Q \times 2^Q, (\{q_0\}, \phi) \text{ if } q_0 \notin G, \Sigma, \delta_m). \\ (\{q_0\}, \{q_0\}) \text{ if } q_0 \in G$$

$$(N \Leftarrow) \quad \delta_m((x, y), a) = \begin{cases} (\delta_s(x, a), \delta_s(y, a) \cup (\delta_s(x, a) \cap G)) & \text{if } x \neq y \\ (\delta_s(x, a), \delta_s(x, a) \cap G) & \text{if } x = y \end{cases}$$

$$G_m = \{(x, x) : x \subseteq Q\}.$$

$$(x_1, x_1) \xrightarrow{u} (x_2, x_2) \Rightarrow \forall q \in x_2 \exists q' \in x_1 \text{ s.t.} \\ q' \xrightarrow{u} q.$$

$$\text{Thus, } L(A_m, G_m) \subseteq L(A) \subseteq L(A_s, G').$$

Lecture 12 (08-09-2021)

08 September 2021 09:47

Determinization

$$A = (\Delta, \Sigma, q_0, \delta, G) \rightarrow_{NBA}$$

Runtree: Given a word α , we have the runtree of A on α which computes all runs of A on α .

$$A_S = (2^{\Delta}, \Sigma, \{q_0\}, \delta_S : 2^{\Delta} \times \Sigma \rightarrow 2^{\Delta}, G' = \{x : x \cap G \neq \emptyset\}).$$

The current state keeps track of the set of reachable states of A .

$$L(A) \subseteq L(A_S)$$

$$A_m = (\{(x, y) \in 2^{\Delta} \times 2^{\Delta} : y \subseteq x\}, \Sigma, \text{initial}, \delta_m, \{(x, x) : x \in \Delta\})$$

$$\delta_m((x, y), a) = \begin{cases} (\delta_s(x, a), \delta_s(y, a) \cup (\delta_s(x, a) \cap G)) & \text{if } x \neq y \\ (\delta_s(x, a), \delta_s(x, a) \cap G) & \text{else} \end{cases}$$

Then, $L(A_m) \subseteq L(A)$.

Recall proof.

$$\text{Let } \alpha = a_0 a_1 a_2 a_3 \dots \in L(A_m)$$

$$s_m = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \xrightarrow{a_0} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \xrightarrow{a_1} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \xrightarrow{a_2} \dots$$

s_m is acc. $\Rightarrow \exists$ inf many i s.t. $x_i = y_i$.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \xrightarrow{u} \begin{pmatrix} x_j \\ y_j \end{pmatrix} \xrightarrow{v} \begin{pmatrix} x_{i+1} \\ x_{i+1} \cap G \end{pmatrix} \xrightarrow{w} \dots$$

$$\forall q \in X_j \exists q' \in X_i \text{ s.t. } q' \xrightarrow[G]{u} q$$

$$\forall q \in X_j \exists q' \in X_i \text{ s.t. } q' \xrightarrow[G]{u} q$$

$$x_0 \xrightarrow{u_0} x_{i_1} \xrightarrow[G]{u_1} x_{i_2} \xrightarrow[G]{u_2} x_{i_3} \xrightarrow[G]{u_3} \dots$$

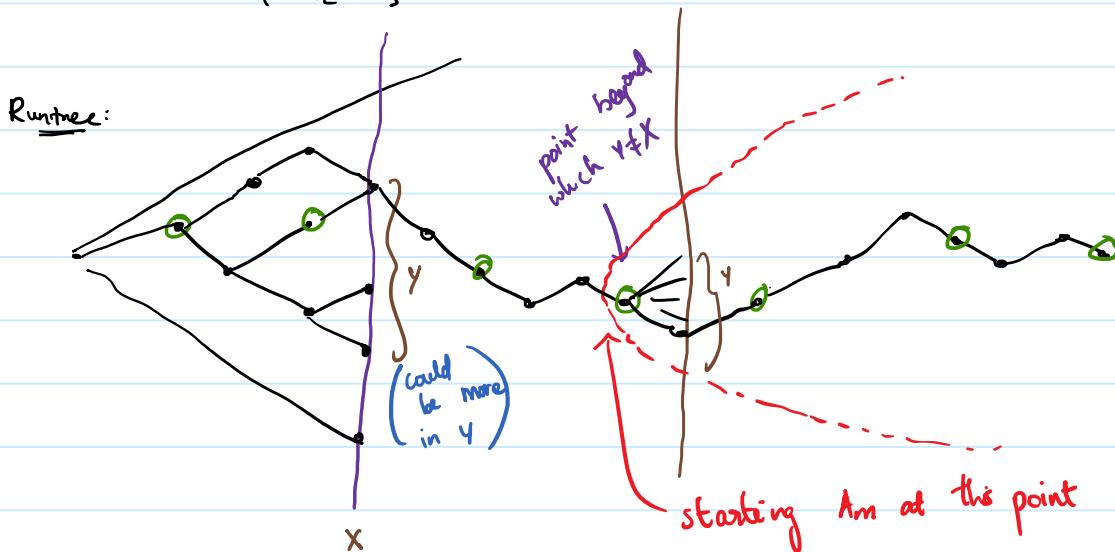
At this point, we applied König's lemma to
see that A has an acc. run on α . Thus, $\alpha \in L(A)$.



- $L(A_m) \subseteq L(A) \subseteq L(A_s)$.

Pick $\alpha \in L(A)$. Suppose $\alpha \notin L(A_m)$.

$$\alpha = a_0 a_1 a_2 a_3 \dots$$



α not acc. by $A_m \Rightarrow$ beyond a point, β_m does not encounter (x, y) .

Trying to show: $\forall \text{ word } \in L(A) \exists \text{ some state s.t. starting } A_m \text{ at that state results in accepting } \alpha$.

Factor α as $u \cdot \beta$ s.t. ' u ' allows A to go to a good state g , and the A_m started at ' g ' accepts β .

Lecture 13 (29-09-2021)

29 September 2021 09:39

Determinisation of BA continued

A - NBA

$$\rightarrow A = (Q, q_0, \Sigma, \Delta, G)$$

A_s - subset automaton

$$\rightarrow G_m = \{x : x \cap G \neq \emptyset\}$$

A_m - marked subset automaton

$$\rightarrow G_m = \{(x, x) : x \in 2^\Sigma\}$$

$$L(A_m) \subseteq L(A) \subseteq L(A_s).$$

If $\alpha \in L(A)$, then \exists a factorisation of α as

$$\alpha = u \cdot \beta$$

and a good state g of A s.t.

$$q_0 \xrightarrow[A]{u} g \quad \text{and} \quad (\{g\}, \{g\}) \xrightarrow[A_m]{\beta} \text{accepts } \beta,$$

i.e., A_m accepts when started at ' g '.

Prop

Let $L = L(A)$ be an ω -regular language. Then, L can be written as

$$L = \bigcup_{i=1}^n U_i \cdot \overrightarrow{V_i}, \quad \text{where } U_i, V_i \text{ are regular languages.}$$

Recall: $\overrightarrow{V} := \lim V := \{\alpha \in \Sigma^\omega : \text{inf many prefixes of } \alpha \text{ are in } V\}$.

Also, if V is accepted by a DFA, then \overrightarrow{V} is accepted by the same automata interpreted as a DBA.

Earlier, we had seen: Every ω -regular L can be written as

$$\bigcup_{\text{finite union}} U \cdot V^\omega$$

for regular languages U and V .

$$L = L(A), \quad A = (\mathcal{Q}, \Sigma, q_0, \Delta, b).$$

$\alpha \in L(A)$, then $\alpha = u \cdot \beta$ where $q_0 \xrightarrow{u} g$
and β is accepted by the A_m started at g .

$$U_g := \{u \in \Sigma^*: q_0 \xrightarrow[A]{u} g\}.$$

$$V_g = \{v \in \Sigma^*: v \text{ is acc. by } A_m \text{ starting at } (f_g, f_g)\}$$

↓
interpreting it as a DFA

Then, $\alpha \in U_g \cdot \overrightarrow{V_g}$.

In fact,

$$L = \bigcup_{g \in G} U_g \cdot \overrightarrow{V_g}.$$

Next class: Construct det Rabin automaton to accept $U \cdot V$.

Exercise: Show that det Rabin ——— are closed under union.

Thus, we conclude that every ω -reg. language can be accepted by a det. Rabin automaton. (In turn, Muller as well.)

Lecture 14 (01-10-2021)

01 October 2021 09:30

Suppose U and V are regular languages with corresponding DFAs given by

$$A_U = (Q_U, q_{\text{0}}^U, \Sigma, \delta_U, F_U),$$

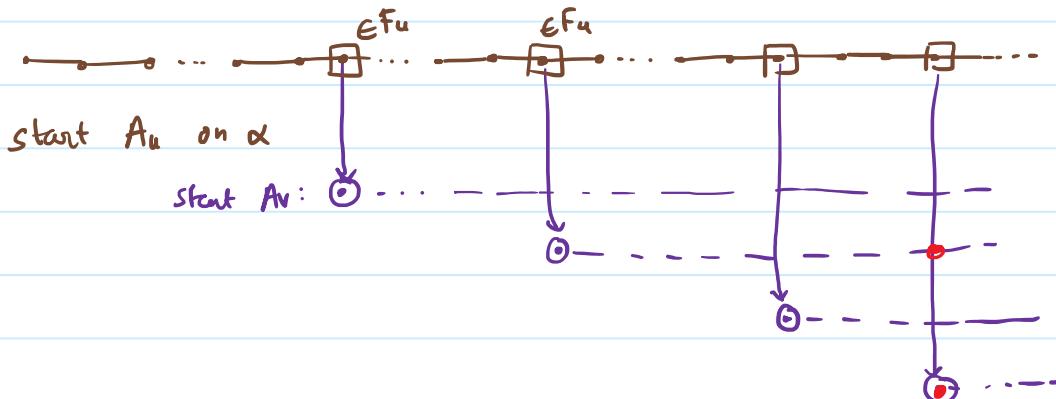
$$A_V = (Q_V, q_{\text{0}}^V, \Sigma, \delta_V, F_V).$$

Goal: To construct a deterministic co-automation (of appropriate kind) which accepts $U \cdot \overrightarrow{V}$ and is closed under finite unions.

Let $\alpha \in U \cdot \overrightarrow{V}$.

Then, we can write $\alpha = u \cdot v$, where $u \in U$ and $v \in \overrightarrow{V}$. i.e., \exists infinitely many prefixes of v in V .

$$\alpha = a_0 a_1 a_2 a_3 a_4 a_5 \dots$$



If we made the guess to jump from A_U at the correct stage, we are done.

Also note: if we are on the red dot on the two A_V s, then the run from that point is identical for both. (Since deterministic)

Thus, we need to run at most $O(|Q_V|)$ copies of A_V at a time.

Here is the automaton:

$$\text{States: } S = Q_u \times (Q_v \times \{\perp\})^{n+1}$$

$$= \{(q_u^0, q_v^1, \dots, q_v^{n+1}) : q_u^0 \in Q_u, q_v^i \in Q_v \cup \{\perp\}\}$$

$q_v^j = \perp$ indicates that no copy of A is running in slot j.

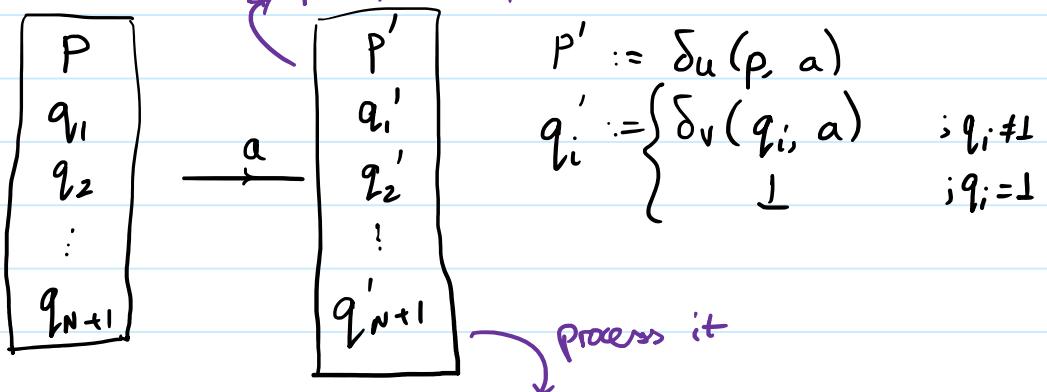
Initial state:

$$S = \begin{cases} (q_0^u, \perp, \perp, \dots, \perp) & \text{if } q_0^u \notin F_u \\ (q_0^u, q_0^v, \perp, \perp, \dots, \perp) & \text{if } q_0^u \in F_u \end{cases}$$

We will maintain invariant that $q_i^v \neq q_j^v$ for $i \neq j$ (unless both are \perp).

In general, $\delta: S \times \Sigma \rightarrow S$ is defined as follows

temporary, not final output



Note that by distinctness, $\exists i \text{ s.t. } q_i = \perp$.

In that case $q'_i = \perp$.

- If $p' \in F_u$, then we redefine $q'_i := q_0^v$.
- Now, we do the "merge"; if there are multiple indices which have ended in same state, change all but lowest index one to \perp .

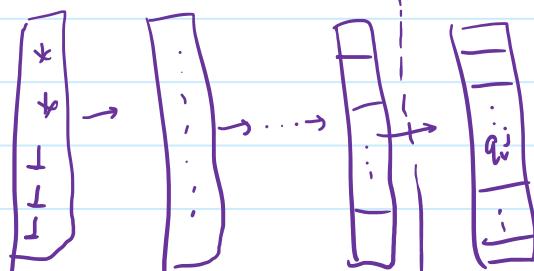
After this processing, whatever we are left with, is the new state.

Suppose $\alpha \in u \cdot \overrightarrow{v}$. Consider the run of

$B = (S, S_0, \Sigma, \delta, *)$ on α .

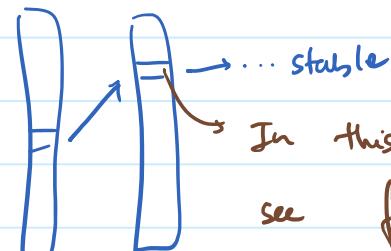
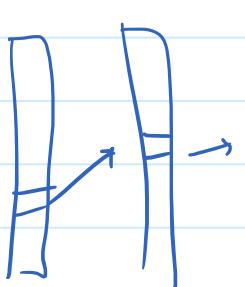
$\sigma = (\omega_1, \omega_2, \dots, \omega_i, \dots)$ on Σ .

$$\alpha = \underbrace{a_0 a_1 a_2 \dots a_k}_{\in U} \mid \underbrace{a_{k+1} a_{k+2} \dots}_{\in V}$$



A copy of a_r is started in slot j

may remain in slot j forever or it may merge with another copy running at a lower indexed slot. And this may happen repeatedly. This jumping cannot happen inf. often since the index strictly decreases. Thus, it will eventually settle at a fixed slot.



In this stable slot, we will see find states of a_r infinitely often in that slot.

Thus, the unique run γ of B on α has the property:

\exists a slot i such that

- 1) the state at slot i will be in F_U inf. often,
- 2) the state at slot i will be \perp finitely often.

Define the sets:

$$E_i = \{ (p, q_1, \dots, q_{i-1}, \perp, q_{i+1}, \dots, q_{n+1}) \in S \}$$

$$= \{ s \in S : \pi_i(s) = \perp \}.$$

$$F_i = \{ s \in S : \pi_i(s) \in F_v \}.$$

$\nearrow i^{\text{th}}$ coordinate

If $\alpha \in U \cdot \vec{V}$, then the unique run ρ has the property that
 $\exists i \text{ s.t.}$

$$\text{Inf}(\rho) \cap E_i = \emptyset \neq \text{Inf}(\rho) \cap F_i.$$

Now, put $\mathcal{L} = \{(E_1, F_1), \dots, (E_{n+1}, F_{n+1})\}$ as the Rabin condition on B .

Our discussion so far has shown: $U \cdot \vec{V} \subseteq L(B)$.

Need to argue the reverse containment.

Let $\alpha \in L(B)$.

$$\alpha = a_0 a_1 a_2 \dots$$

$$\rho = s_0 s_1 s_2 \dots$$

Since ρ satisfies the Rabin condition, $\exists i \text{ s.t. } \text{Inf}(\rho) \cap E_i = \emptyset$
 $\& \text{Inf}(\rho) \cap F_i \neq \emptyset$.

Since $\text{Inf}(\rho) \cap E_i = \emptyset$, \exists a point at which i is started and never stopped. The prefix until that would've been in U and the suffix from there is in \vec{V} since $\text{Inf}(\rho) \cap F_i \neq \emptyset$.

McNaughton's Theorem:

Every ω -reg lang. can be accepted by a Muller automaton.

We showed for Rabin. Can do from Rabin to Muller without changing states and transitions.

Thm.

Every ω -regular language is a boolean combination of languages accepted by DBA (or equivalently, languages of the form \vec{U}).

$\nearrow \wedge, \vee, \neg$

Brek. Let $A = (Q, q_0, \Sigma, \delta, F = \{F_1, \dots, F_s\})$ be a det Muller automaton.

$$L(A) = \bigcup_{i=1}^s L((Q, q_0, \Sigma, \delta, \{F_i\})).$$

Thus, wlog assume $|F| = 1$. Write $F = \{F\}$.

$\forall q \in Q$, define $L_q = \left\{ \alpha : \text{unique run of } A \text{ on } \alpha \text{ visits } q \text{ inf. often} \right\}$
 $= \vec{U}_q$, where U_q is the lang. accepted with $\{q\}$ as final set.

Since we are using Muller condition, we have

$$L(A) = \left(\bigcap_{q \in F} \vec{U}_q \right) \cap \left(\bigcap_{q \notin F} \neg(\vec{U}_q) \right). \quad \text{B}$$

Lecture 15 (08-10-2021)

08 October 2021 10:03

Safra's determinisation construction

Given: NBA : $A = (Q, q_0, \Sigma, \delta, G)$.

Goal: Construct a det. Rabin automaton which accepts $L(A)$.

High level: Make a tree with nodes running copies of the marked subset automaton.

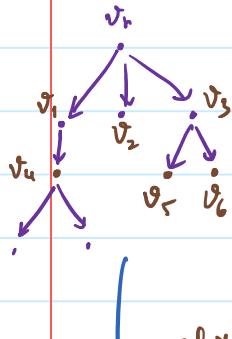
Notation: A tree $T = (V, v_r, \pi)$ where

$V \rightarrow$ a set of vertices,

$v_r \rightarrow$ a special vertex of V , the root of T ,

$\pi : V \setminus \{v_r\} \rightarrow V$, $\pi(v)$ is the parent of v ,

$\text{children}(v) := \{v' \in V : \pi(v') = v\} = \pi^{-1}\{v\}$.



For each node (vertex) v of T , there is an ordering (visually left to right) on $\text{children}(v)$.

$\text{children}(v) = (v_1, v_2, v_3)$. v_4 is to the left of v_6 .
 "eldest" "youngest"

Notation: $n := |Q|$, $\mathcal{L} := \{\ell_1, \dots, \ell_{2n}\}$.
 ↪ set of labels

Def'n: A Safra tree is $s = (T, \sigma, \chi, \gamma)$, where

- $T = (V, v_r, \pi)$ is an ordered tree,
- $\sigma : V \longrightarrow 2^Q \setminus \{\emptyset\}$ has the following properties:
 let $v \in V$ with $\text{children}(v) = \{v_1, \dots, v_k\}$.

Then,

$$\sigma(v) \supseteq \sigma(v_1) \sqcup \dots \sqcup \sigma(v_k).$$

+

↳ disjoint

That is, if $v \neq v'$ with $\pi(v) = \pi(v')$, then
 $\sigma(v) \cap \sigma(v') = \emptyset$.

- $\chi : V \longrightarrow \{\text{white, green}\}$

- $\lambda : V \hookrightarrow \mathcal{L}$ is injective.

Different vertices are assigned different labels.

Note: The above implicitly forces $|V| \leq 2n$. In fact, now is true.

Claim. Let $S = (T, \sigma, \chi, \lambda)$ be a Safr tree with $T = (V, V_r, \pi)$. Then, $|V| \leq n$.

Proof. Claim : $|V| \leq |\sigma(V_r)|$.

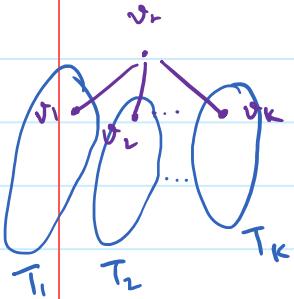
Proof. Will prove this by induction on $|\sigma(V_r)|$.

Base case is clear.

Each T_i is a Safr tree.

$$\begin{aligned}
 |V| &= 1 + \sum |T_i| \\
 &\leq 1 + \sum |\sigma(V_i)| \\
 &\leq 1 + (|\sigma(V_r)| - 1) \\
 &= |\sigma(V_r)|
 \end{aligned}$$

By induction, since
 $\sigma(V_i) \subsetneq \sigma(V_r)$
 $\bigcup \sigma(V_i) \subsetneq \sigma(V_r)$



Since $\sigma(V_r) \subseteq Q$, it follows that $|V| \leq |\sigma(V_r)| \leq n$. \square

Let us fix a state: Safr tree $S = (T = (V, V_r, \pi), \sigma, \chi, \lambda)$.

On reading a, $S \xrightarrow{a} S'$, where S' is to be defined:

- Expand $T \rightarrow T_i = (V_i, \sigma_i, \pi_i)$ as follows:
for each $v \in V$, if $\sigma(v) \cap G \neq \emptyset$, then
we create a new node v' which is the
youngest (rightmost) child of v .

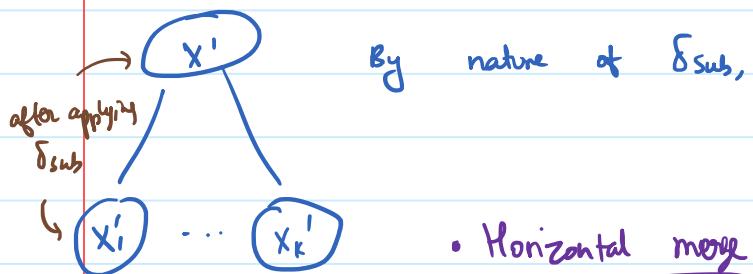
- Define σ'_i of such a v' as
 $\sigma'_i(v') = \sigma(v) \cap G$.

- Note that the tree T_i can have at most $2n$ nodes. We can use labels in $\mathcal{L} \setminus \lambda(V)$ to
label the vertices in $V_i \setminus V$.
Call this labelling λ_i .
 $(\lambda_i|_V = \lambda)$

- Now, we update the subsets using the subset automaton.

$$\sigma'_i(v) := \delta_{\text{sub}}(\sigma_i(v), a).$$

This intermediate labelled tree $(T_i, \sigma'_i, \lambda_i)$ is now converted into a valid Safra tree.



By nature of δ_{sub} , we have $x'_i \subseteq x'$
 $\forall i$.

- Horizontal merge

If q' is present in multiple sets, then keep it only in its leftmost vertex.

- Vertical merge

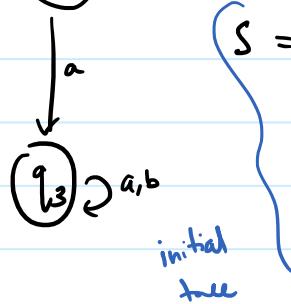
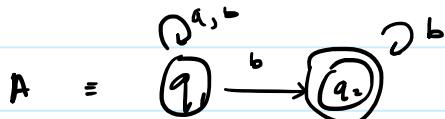
If $v = \bigcup_{w \in \text{children}(v)} \sigma(w)$, delete all children
of v .

. Colouring : if there was a vertical merge at v ,
colour it green
white otherwise

Lecture 16 (13-10-2021)

13 October 2021 10:02

Example.



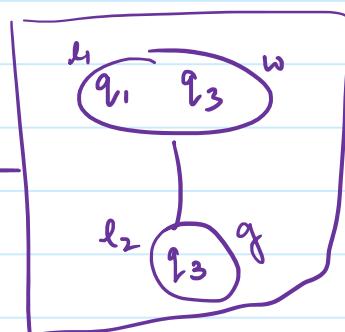
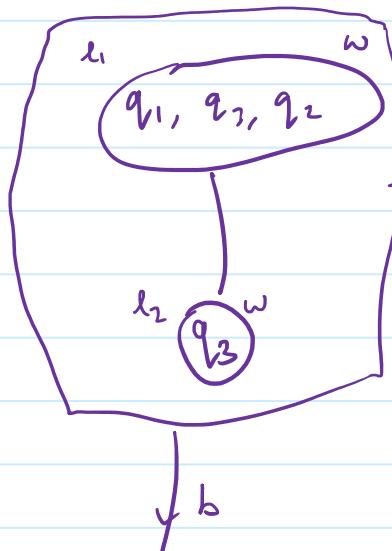
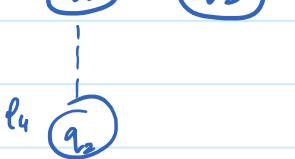
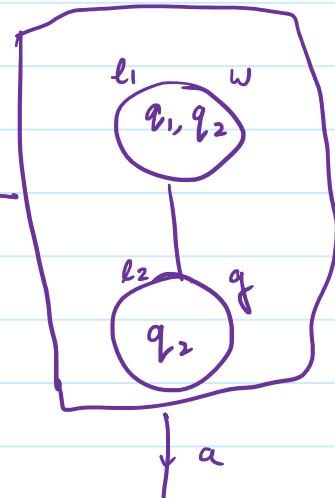
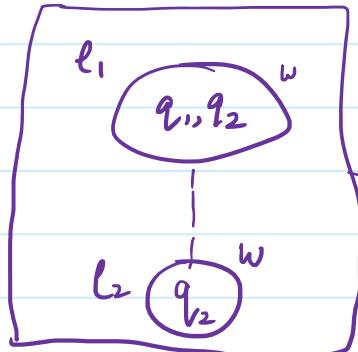
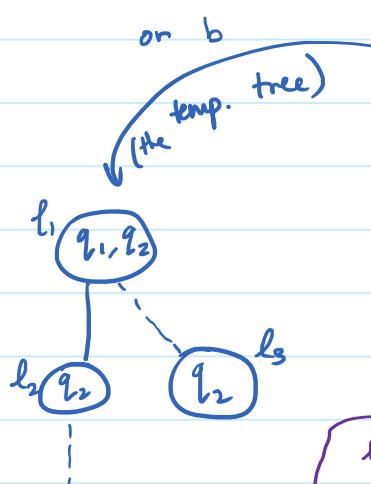
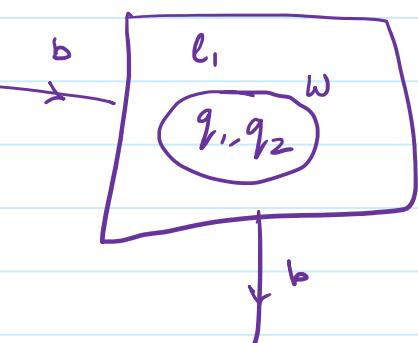
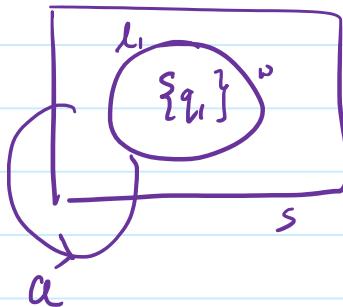
$$L = \{l_1, \dots, l_6\}$$

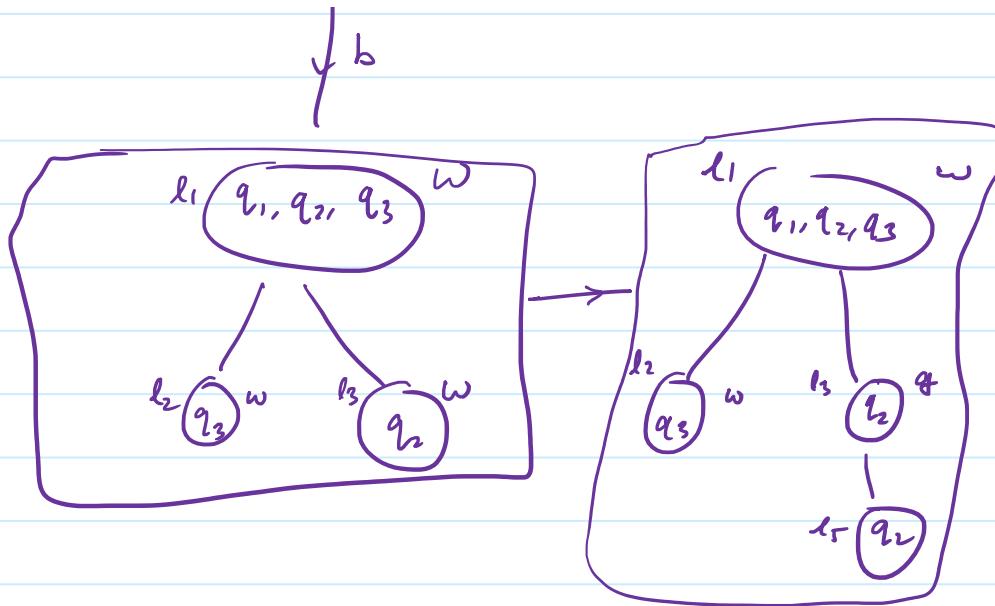
$$S = (T, \sigma, x, \lambda), T \in (V, V_r, \pi)$$

$$\sigma : v \rightarrow 2^A$$

$$x : v \rightarrow \{w, g\}$$

$$\lambda : v \rightarrow L$$





$$A = (Q, q_0, \Sigma, \delta, \mathcal{G})$$

Büchi Automaton

$$B = (S, s_0, \Sigma, \Delta \text{ Rabin condition})$$

$$\Delta = \{(E_1, F_1), \dots, (E_{2n}, F_{2n})\}.$$

$E_i = \{s \in S : s \text{ does not contain a node of label } 'l_i'\}$,

$F_i = \{s \in S : s \text{ has a node } v \text{ s.t. } \gamma(v) = l_i \text{ and } \chi(v) = \text{green}\}$.

Theorem. $L(A) = L(B)$.

Proof. Let $\alpha \in \Sigma^\omega$. Suppose β is the unique run of B on α .

$$\beta(i) = s_i = (T_i, \sigma_i, \chi_i, \gamma_i).$$

Further, let $j < k$, and $l \in \mathcal{L}$ be s.t.

(i) for all positions $i \in \{j, \dots, k\}$, the tree T_i has a node labelled l , say "v".

(2) $\chi_j(v) = \chi_k(v) = \text{green}$
and $\forall i \in \{j+1, \dots, k-1\} : \quad \chi_i(v) = \text{white}.$

Lecture 17 (22-10-2021)

22 October 2021 09:33

Complexity of Safra's Determinisation

Let $A = (Q, q_0, \Sigma, \Delta, G)$ be an NBA on $n := |Q|$ states.
 $B = (S, s_0, \Sigma, \delta, \Omega)$ det. Rabin automaton obtained
from Safra's construction.

$$S = (\mathcal{T} = (V, V_r, \pi), \sigma : V \rightarrow 2^Q, x : V \rightarrow \{\omega, g\}, \lambda : V \rightarrow \mathcal{L}).$$

Question : $|S| = ?$

Lecture 18 (27-10-2021)

27 October 2021 10:11

Linear temporal logic (LTL)

Atomic propositions: Abstract propositions (can be true or false)
Truth value can change with time

$$\mathcal{P} = \{P_0, P_1, P_2, \dots\}.$$

LTL - formulae over \mathcal{P} .

Syntax: $\Phi \equiv \mathcal{P}^{\mathcal{P}}$

temporal modalities

$$\Phi \equiv p \mid \neg \alpha \vee \beta \mid \alpha \wedge \beta \mid \Box \alpha \mid \alpha \wedge \Diamond \beta$$

"next" α "until" β

Defn. A model M is a function

$$M: \mathbb{N} \rightarrow 2^{\mathcal{P}}.$$

That is, M can be thought of as a sequence

$$M = P_0, P_1, P_2, \dots \quad \text{with } P_i \subseteq \mathcal{P} \quad \forall i.$$

Interpret: $P_i =$ set of those atomic propositions which are true at time i .

Example. $\mathcal{P} = \{p, q\}$

$$M = \emptyset, \{p\}, \{p, q\}, \emptyset, \{q\}, \dots$$

\downarrow \downarrow \dots
 p true q false
 p, q both false

Semantics:

Let M be a model, $i \in \mathbb{N}$, α is an LTL formula.

" $M, i \models \alpha$ " \equiv α is true in M at time i .

As usual, we formally define the above by structural induction:

- $\alpha = p$ (for some $p \in P$).

" $M, i \models p$ " \equiv $p \in M(i)$.

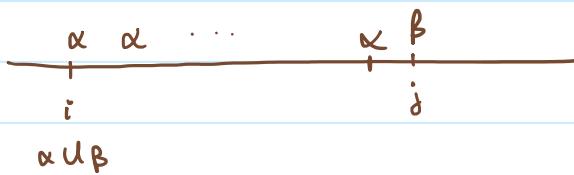
- " $M, i \models \neg \alpha$ " $\equiv \neg (M, i \models \alpha)$

\hookrightarrow also written $M, i \not\models \alpha$

- \wedge, \vee also done as expected. (Can derive \Rightarrow , etc.)

- " $M, i \models O\alpha$ " $\equiv M, i+1 \models \alpha$.

- " $M, i \models \alpha \sqcup \beta$ " $\equiv \exists j \geq i \left[M, j \models \beta \text{ and } \forall k : i \leq k \leq j \Rightarrow M, k \models \alpha \right]$



Note: if β is true at time i itself, then done directly.

Def.: An LTL-formula α is **satisfiable** if \exists a model M and (time instance) $i \in \mathbb{N}$ s.t. $M, i \models \alpha$.

Obs.: ① α is satisfiable iff \exists a model M' s.t. $M', 0 \models \alpha$.

(\Leftarrow) Truncate the appropriate model M at i . Since LTL only has future modalities, we are done.)

② $Voc(\alpha) = \text{set of propositions mentioned in } \alpha$.

The truth value of α in a model M depends only on $Voc(\alpha)$.

$M : D \quad P \quad P_1 \quad D \quad \dots$

$Voc(\alpha) - \{n\}$

Inductively define

Given a formula α , we have

$$L(\alpha) = \{ M \in \left[2^{\aleph_0c(\alpha)}\right]^\omega : M, 0 \models \alpha \}.$$

We will show that $\mathcal{L}(\alpha)$ is regular by converting α to a BA A_α s.t. $L(A_\alpha) = \mathcal{L}(\alpha)$, i.e., given $M = P_0 P_1 P_2 \dots$, with $P_i \subseteq \text{Voc}(\alpha)$, we have $M, \sigma \vdash \alpha \iff A_\alpha \text{ accepts } M$.

Start of automata theoretic approach to model-checking.

Vardi - Wolper construction : $\alpha \leadsto \beta$

Key idea : to keep track of truth values of all subformulae of α .

The closure $CL'(\alpha)$ is the smallest set of formulas s.t.

- 1) $\alpha \in CL'(\alpha)$,
 - 2) $\neg\beta \in CL'(\alpha) \Rightarrow \beta \in CL'(\alpha)$,
 - 3) $\beta \vee \gamma \in CL'(\alpha) \Rightarrow \beta, \gamma \in CL'(\alpha)$,
 - 4) $O\beta \in CL'(\alpha) \Rightarrow \beta \in CL'(\alpha)$
 - 5) $\beta \vee \gamma \in CL'(\alpha) \Rightarrow \beta, \gamma, O(\beta \vee \gamma) \in CL'(\alpha)$

Example: $\alpha = p \vee q$; $CL'(\alpha) = \{ p \vee q, p, q, o(p \vee q) \}$

$$(1(x)) \in \mathcal{G}'(x) \cup \{\exists y : y \in \mathcal{G}'(x)\}$$

$$CL(\alpha) = CL'(\alpha) \cup \{\neg\beta : \beta \in CL'(\alpha)\}.$$

Here, we identify $\neg\neg\beta$ with β

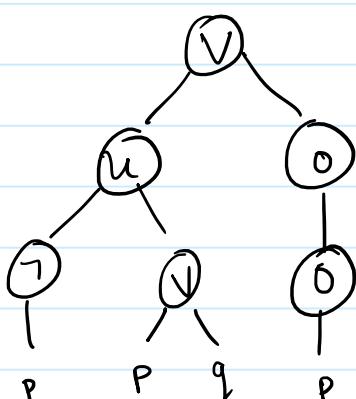
$$CL(\alpha) = \{ p \vee q, \neg(p \vee q), \neg\neg(p \vee q), \neg(p \vee q), \neg p, \neg q, \neg\neg(p \vee q) \}.$$

$$CL(\neg(p \vee q)) = \{ \neg(p \vee q), p \vee q, \neg p, \neg q, \neg\neg(p \vee q) \}.$$

Claim: $|CL(\alpha)|$ is $\Theta(|\alpha|)$.

↳ big-O notation (not "next")

$|\alpha| \rightarrow$ size of parse tree



$$(\neg p \vee (\neg p \vee q)) \vee o(p)$$

Each internal node gives a subformula and viceversa.

$$|CL'(\alpha)| \leq 2|\alpha|,$$

$$|CL(\alpha)| \leq 4|\alpha|.$$

Defn.: An atom A is a subset of $CL(\alpha)$ s.t.

i) $\forall \beta \in CL(\alpha), \neg\beta \in A$ iff $\beta \notin A$.

ii) $\forall \beta \vee \gamma \in CL(\alpha), \beta \vee \gamma \in A$ iff $\beta \in A$ or $\gamma \in A$.

iii) $\forall \beta \wedge \gamma \in CL(\alpha), \beta \wedge \gamma \in A$ iff $\beta \in A$ and $\gamma \in A$.

iii) $\forall \beta \wedge \gamma \in CL(\alpha), \beta \wedge \gamma \in A$ iff $\beta \in A$ and $\gamma \in A$.

no need if we

remove \wedge as a base

Connective $(p \wedge q = \neg(\neg p \vee \neg q))$

AT - set of all atoms.

$$A_\alpha = (S = AT, \rightarrow \subseteq S \times 2^{\text{Var}(\alpha)} \times \subseteq S_{in}, \dots)$$

Suppose $A, B \in AT$ and $P \subseteq \text{Var}(\alpha)$:

$A \xrightarrow{P} B$ if

- 1) $A \cap \text{Var}(\alpha) = P$,
- 2) $\forall \alpha_B \in CL(\alpha), \alpha_B \in A \Leftrightarrow \beta \in B$

Lecture 19 (29-10-2021)

29 October 2021 09:30

AT - Set of all atoms

$$A_\alpha = (S = AT, S_{in}, \rightarrow \subseteq S \times 2^{\text{Voc}(\alpha)} \times S, Acc)$$

$$S_{in} = \{ A \in AT \mid \alpha \in A \}$$

$A, B \in S, P \subseteq \text{Voc}(\alpha) :$

$$A \xrightarrow{P} B \text{ iff } \begin{aligned} (1) \quad & A \cap \text{Voc}(\alpha) = P, \text{ and} \\ (2) \quad & \forall O_B \in CL(\alpha), O_B \in A \Leftrightarrow \beta \in B. \end{aligned}$$

Acc. : Generalised Büchi automaton

$$G_1, \dots, G_k \subseteq Q$$

$$\left(\begin{array}{l} \text{Ex :} \\ G_{BA} = BA \end{array} \right)$$

P is acc. if $G_i \cap P \neq \emptyset \quad \forall i \in \{1, \dots, k\}$.

Here, we define the collections as follows:

Let $\beta, \gamma_1, \dots, \beta_k, \gamma_k$ be all the until formulae in $CL(\alpha)$.

Then, define

$$G_i := \{ A \mid \beta_i \mathbin{\parallel} \gamma_i \notin A \text{ or } \gamma_i \in A \}$$

$$= \{ A \mid \beta_i \mathbin{\parallel} \gamma_i \in A \Rightarrow \gamma_i \in A \}$$

Hm.

$$L(A_\alpha) = \{ M \in P, P_1, P_2, \dots \in (2^{\text{Voc}(\alpha)})^\omega \mid M, O \models \alpha \}$$

$$= \mathcal{L}(\alpha).$$

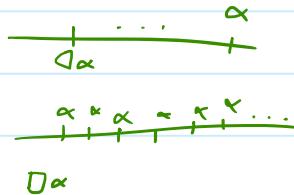
Lecture 20 (03-11-2021)

03 November 2021 10:07

Derived temporal modalities:

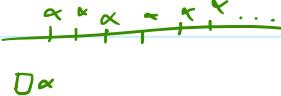
$$\textcircled{1} \quad \diamond \alpha \equiv \text{true} \wedge \alpha$$

\equiv eventually α

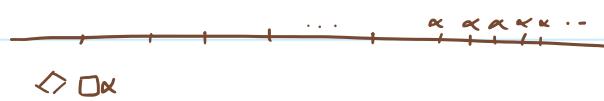


$$\textcircled{2} \quad \Box \alpha \equiv \neg \diamond \neg \alpha$$

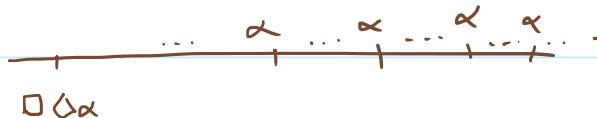
\equiv henceforth α



Example ① $\diamond \Box \alpha = \alpha$ is a stable property



$$\textcircled{2} \quad \Box \diamond \alpha = \alpha \text{ inf. often}$$



③ Suppose K processes are competing for a shared resource.

r_i — i^{th} process requests access to resource by setting
 r_i to TRUE. (Else it is FALSE.)

g_i — the i^{th} process has been granted the resource.

$$\cdot \bigwedge_{i=1}^K \Box(r_i \Rightarrow \diamond g_i)$$

↳ every request is eventually fulfilled.

$$\cdot \bigwedge_{i=1}^K \Box(r_i \Rightarrow r_i \wedge g_i)$$

↳ r_i remains true until granted