

# Lecture 1 (30-07-2021)

30 July 2021 09:28

## Automata on infinite words

### Some notations

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

(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  $\Sigma$ )

(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  $\alpha = a b b a b b a b b$

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

$\alpha = (abb)^\omega$

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

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

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

$\gamma$  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  $\Sigma^\omega$  is not a countable set.  
Or,  $\Sigma^*$  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**  $\rho$  of  $A$  on  $\alpha$  is an infinite sequence of states

(run)

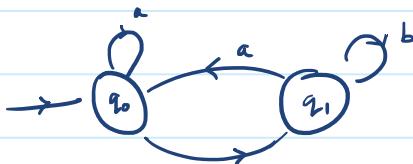
$$\rho = q_0 q_1 q_2 \dots$$

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  
 $\rho : \omega \rightarrow Q$  s.t.  $\rho(0) = q_0$  and  $(\rho(n), \alpha(n), \rho(n+1)) \in \Delta \quad \forall n \in \omega$ .

Example.



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

$$\rho = a \cdot (a \cdot a)^\omega$$

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

"Acceptance condition": (Büchi automata)

$\alpha \rightarrow$  input word

$f \rightarrow$  a run of  $A$  on  $\alpha$

$\text{Inf}(f) :=$  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 states".

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

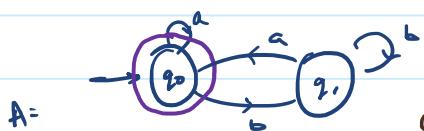
(Thus, some good state appears infinitely often.)

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

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

↳ Language of  $A$

Example



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

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

many 'a' g.

Prof. let the right side be L.

- $L(A) \subseteq L$ :

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

Note that A is deterministic, thus  $\alpha$  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  $\alpha$  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  $\epsilon$  transitions in this course.

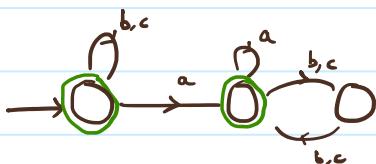
Fact. Even though we insisted on single initial state, the expressive power does not change if we allow more.  
(It is simply for convenience.)

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

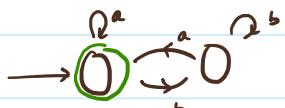
$L = \text{every 'a' is eventually followed by a 'b'}$



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

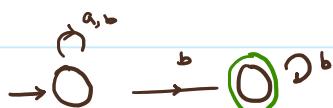


(3)  $\Sigma = \{a, b\}$ ,  $L = \text{inf. many 'a's}$



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

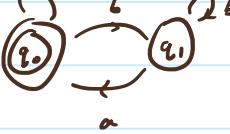
Q. What is a BA for  $\bar{L}$ ?



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

Note that we had an DBA for  $L$ : 

A Deterministic Büchi Automaton (DBA) with two states,  $q_0$  and  $q_1$ . The start state is  $q_0$ , which has a self-loop labeled 'a,b'. There is a transition from  $q_0$  to  $q_1$  labeled 'b'. State  $q_1$  has a self-loop labeled 'b'.

Note that we had an DBA for  $L$  :  $\rightarrow$  

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

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.

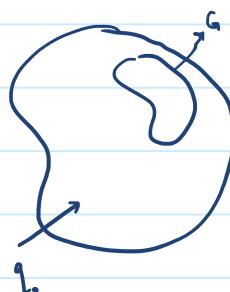
For the sake of contradiction, assume that  $\exists$  DBA  $A$  such that  $L(A) = \bar{L}$ .

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  $\alpha_0$ .



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

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

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

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

$\alpha_1 = b \dots b^{n_1} a b b \dots$   
 $\beta_1 = \square \dots \square \xrightarrow{\text{EG}} \dots \square$ 
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  $\beta_2$  matches  $\beta_1$  until  $b^{n_1} a b^{n_2} a$ .

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

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

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  $\omega$ -regular if there exists a (possibly non-deterministic) Büchi automaton  $A$  such that  $L(A) = L$ .

## CLOSURE PROPERTIES OF $\omega$ -REGULAR LANGUAGES

i) closure under union:

$$L_1 = L(A_1), \quad A_1 = (Q_1, q_0^1, \Sigma, \Delta_1, G_1),$$

$$L_2 = L(A_2), \quad A_2 = (Q_2, q_0^2, \Sigma, \Delta_2, G_2).$$

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

We do the usual product construction.

$$(Q_1 \times Q_2, (q_0^1, q_0^2), \Sigma, \Delta, \underbrace{G_1 \times G_2 \cup Q_1 \times G_2}_{\delta})$$

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

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

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

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

$$s^2 = q_0^2 q_1^2 q_2^2 \dots \quad \text{---} \quad A_2 \quad \text{---}$$

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

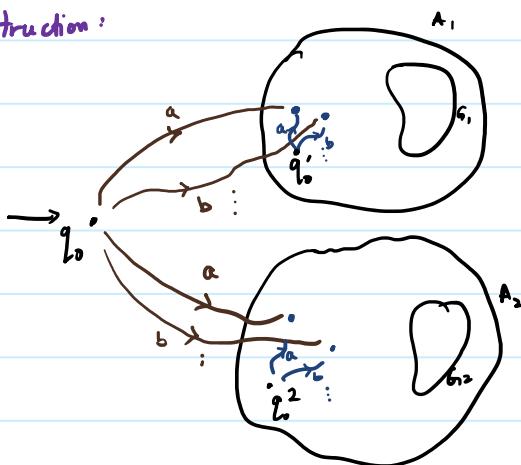
Here we assume that each  $\alpha \in \Sigma^\omega$  has at least one

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

With the above assumption,

$G = (G_1 \times Q_2) \cup (Q_1 \times G_2)$  gives  
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

$$G = G_1 \times G_2.$$

$$A = A_1 \times A_2.$$

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

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

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

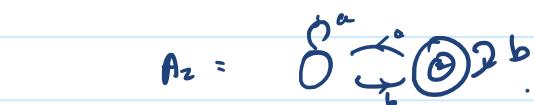
Then there are accepting runs  $p_i$  on  $A_i$ .

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

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:



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

Doesn't work! Slightly modified.

$$Q = Q_1 \times Q_2 \times \{1, 2\},$$

↑ indicates the component being  
 "searched" for a good state

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

$$\Delta = (q_1, q_2, 1) \xrightarrow{a} (q_1^1, q_2^2, 1) \text{ if } \begin{cases} q_1 \xrightarrow{a} q_1 \\ q_2 \xrightarrow{a} q_2 \end{cases}$$

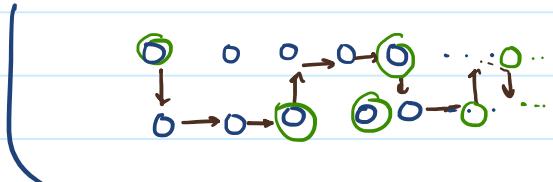
$q_1 \notin G_1$

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

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 under projection:

$\pi: \Sigma_1 \times \Sigma_2 \rightarrow \Sigma_1$  induces  $\pi: (\Sigma_1 \times \Sigma_2)^\omega \rightarrow \Sigma_1^\omega$

$$\begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \dots \mapsto a_0 a_1 a_2 \dots$$

If  $L \subseteq (\Sigma_1 \times \Sigma_2)^\omega$  is  $\omega$ -regular, so is  $\pi(L)$ .

Let  $A = (Q, q_0, \Sigma_1 \times \Sigma_2, \Delta, G)$  be a BA with  $L(A) = L$ .

Goal: Construct  $B$  s.t.  $L(B) = \pi(L)$ .

Define  $B = (Q, q_0, \Sigma_1, \Delta', G)$ , where

$$\Delta' = \left\{ q \xrightarrow{a} q': \exists b \in \Sigma_2 \text{ s.t. } q \xrightarrow{(a,b)} q' \text{ in } \Delta \right\}.$$

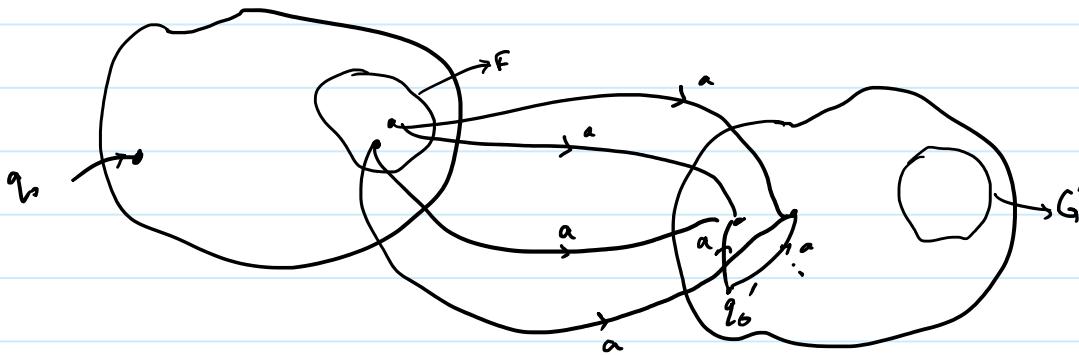
(Take original automata and erase all the second component.)

$$U \subseteq \Sigma^*, L \subseteq \Sigma^\omega, U \cdot L \subseteq \Sigma^\omega.$$

(The concat. of a finite word followed by an infinite word is defined in the natural way.)

(Nonsense : Concat :  $\Sigma^\omega \times \Sigma^* \rightarrow \Sigma^*$  or concat :  $\Sigma^* \times \Sigma^\omega \rightarrow \Sigma^\omega$ )

Closure :  $U \subseteq \Sigma^*$  regular  $A = (Q_0, q_0, \Sigma, \Delta, F)$ ,  
 $L \subseteq \Sigma^\omega$   $\omega$ -regular  $B = (Q'_0, q'_0, \Sigma', \Delta', G)$ .  
 $\xrightarrow{NFA, L(A) = U}$   
 $\xleftarrow{BA, L(B) = L}$



Keep them disjoint and all possible transitions  
of the form:

$$q_f \xrightarrow{a} q' \quad \text{where } q_f \in F \quad \text{and } q'_0 \xrightarrow{a} q' \text{ in } \Delta'$$

Keep  $G$  as  $G'$

\*

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

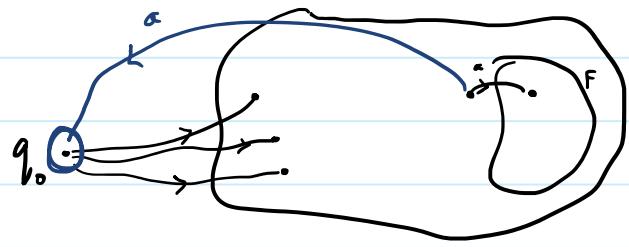
$U^\omega = \{ \alpha \in \Sigma^\omega : \alpha \text{ has a factorisation of the form}$

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

Closure: If  $U \subseteq \Sigma^*$  is regular, then  $U^\omega$  is  $\omega$ -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  $\omega$ -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  $\omega$ -regular.)

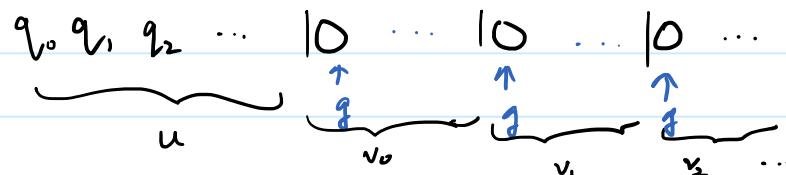
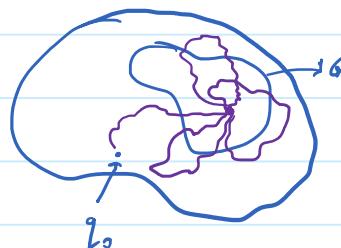
Proof: Let  $A$  be a BA s.t.  $L(A) = L$ .

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

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

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

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  $\omega$ -regular.

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

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

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

### Observations:

(i)  $\sim_A$  is an equivalence relation on  $\Sigma^*$

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

(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.

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

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

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

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

Then, we have a run

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

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

so does  $\beta_B$ .

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

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

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,  $x_\infty$  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.  $\sum^\omega = \bigcup_{i,j} U_i \cdot U_j^\omega$ .

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

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

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

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

Define the following  $\chi_\alpha$  on  $(\mathbb{N}, \leq)$  as:

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

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

*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.  $\square$

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

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

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  $\omega$ -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 -  $a(x), b(x), \dots$   $\rightarrow$  unary predicate  
for  $a, b, \dots \in \Sigma$   
and  $x$  is a Fo variable.

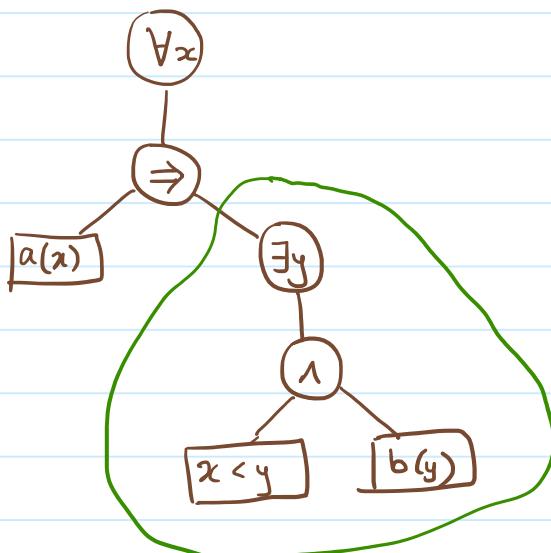
$x < y \rightarrow$  binary predicate

### 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:

$\varphi(x_1, \dots, x_m) \equiv \varphi$  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  $\varphi_2 = \exists x \forall y [(x < y) \Rightarrow \neg a(y)].$

If  $w$  is finite, then it will satisfy  $\varphi_2$ .  
 $(\neq \epsilon)$

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  $\omega$ -regular languages

Monadic Second-Order Logic over words

Extends FO - 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  
 $\equiv \forall y (x = y \vee x < y)$ .

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

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

$x \neq y$  and  
 $x < y \Leftrightarrow$  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 ' $'$ ' in  $M\sigma$ .

Defn. Let  $\varphi$  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  $\omega$ -regular.

# Lecture 7 (20-08-2021)

20 August 2021 09:37

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

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

Proof. ( $\Rightarrow$ ) Suppose  $L$  is  $\omega$ -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  $\varphi_A$  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  $\alpha$  via an accepting run  $\rho$ .

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

$\forall q \in Q, X_q \equiv$  the set of positions in  $\alpha$  when  
this run  $\rho$  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  $Q = \{0, 1, \dots, m\}$ .

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

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

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

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

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

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

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

( $\Leftarrow$ ) Given: MSO sentence  $\varphi$ .

Goal: Construct BA A s.t.

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

As in the finite case, we use MSO<sub>0</sub>-logic

$\hookrightarrow$  substitute position variables by singleton set vars.  
 $\hookrightarrow$  more atomic predicates:  $\text{Sing}(x)$ ,  $a(x)$ ,  $S(x, y)$ ,  $x \leq y$

$\downarrow$  singleton  
 $\downarrow$   $x, y$  are sing and the single positions are related by  $S$

these can be defined in MSO.

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 MSO<sub>0</sub> formula, construct a BA A s.t.  $L(A) = L(\varphi)$ .

The construction of A proceeds by structural induction  $\varphi$ .

In fact: let  $\varphi(x_1, \dots, x_n)$  be an MSO<sub>0</sub>-formula with free  
(set-) variables  $x_1, \dots, x_n$

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

$$\begin{array}{ll} \alpha = & a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ \dots \\ p_1 = \{0, 1, 4, \dots\} & | \quad 0 \quad 0 \quad 1 \quad 1 \quad \dots \\ p_2 = \emptyset & 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \\ \vdots & \vdots \\ p_n = \{0, 1, n, \dots\} & | \quad 0 \quad 1 \quad 0 \quad 1 \quad \dots \end{array} \quad \left. \begin{array}{l} \text{characteristic} \\ \text{vectors} \end{array} \right\}$$

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

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

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

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

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

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

base.  $\varphi$  - atomic predicate

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

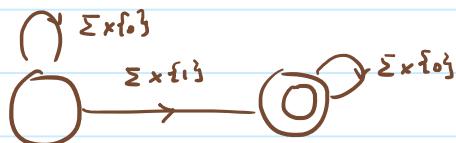
$$\alpha' = \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \dots$$

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

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

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

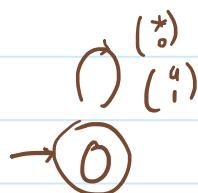
$$\begin{array}{l} a_i \in \Sigma \\ b_i \in \{0,1\} \end{array}$$



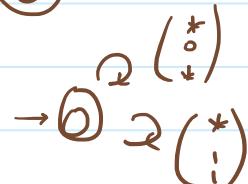
$$\varphi = S(x_1, x_2) \quad ; \quad \Sigma_2 = \Sigma x \{f_0, 1\} \wedge f_0, 1\}$$



$$\varphi = a(x_1)$$



$$\varphi = x_1 \subseteq x_2$$



Inductive step.

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

$$\text{Free}(\varphi_1) \subseteq \{x_1, \dots, x_n\}.$$

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

By induction, we have appropriate automata  $A_{\varphi_i}$  for  $\varphi_i$ .

But the alphabet for both is same. Can take union of  $S\Delta$ .

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

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

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

Note :  $\alpha' \models \varphi \Leftrightarrow \exists \text{ a set } P_n \text{ s.t.}$

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

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  $\omega$ -regular language is  $\omega$ -regular.

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

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

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

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  $\omega$ -words but was not so clear how to extend regularity.

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

Given an MSO sentence  $\varphi$ , we can check if there

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

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

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

and we can produce the above  $u, v$ .

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

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

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

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

singly exp

doubly exp, ...,  $k$ -ary exp, ..

$\nwarrow$  can we do better for satisfiability?

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

Note:  $\psi \rightsquigarrow A_\psi \rightsquigarrow \psi_{A_\psi}$

$\hookrightarrow$  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:

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

Given a sentence  $\varphi$ , we would like to know if  $\varphi$  is true.

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

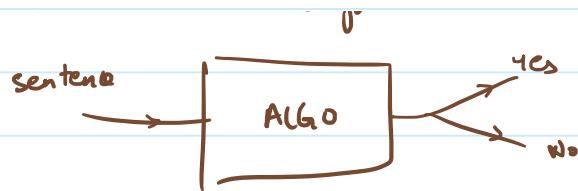
? what does this mean?

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  $\{0, 1\}^n$ .

$\rightarrow$  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)

M605:

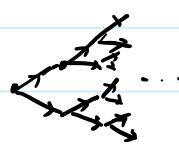
[W]S1S : Subsets are only allowed to be finite subsets  
1 weak

Büchi showed that S1S is decidable

S2S is decidable

147ds  
(Rabin's theorem)

too successors



MONA : Logic  $\rightarrow$  Automata

New modes of acceptance for automata on infinite words

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

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

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

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

$\downarrow$   
"states at  $\infty$ "

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

$p$  is accepting if

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

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

Streett : Dual to Rabin.

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

$p$  is accepting if

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

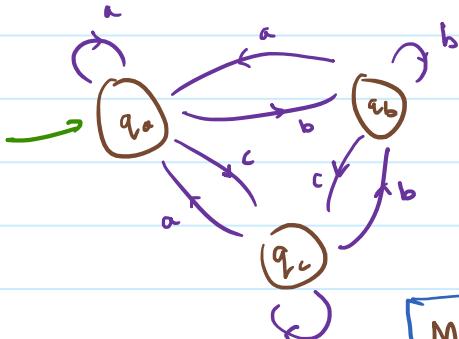
$\uparrow$   
equivalently

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

$\hookrightarrow$  If  $F_i$  is visited infinitely often, then  
 $\therefore E_i$ .

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

$L = \{\alpha \in \Sigma^\omega \mid \text{if 'a' occurs inf. often in } \alpha, \text{ then 'b' does not}\}$ .



Street - Condition:

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

Rabin - Condition

Note:

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

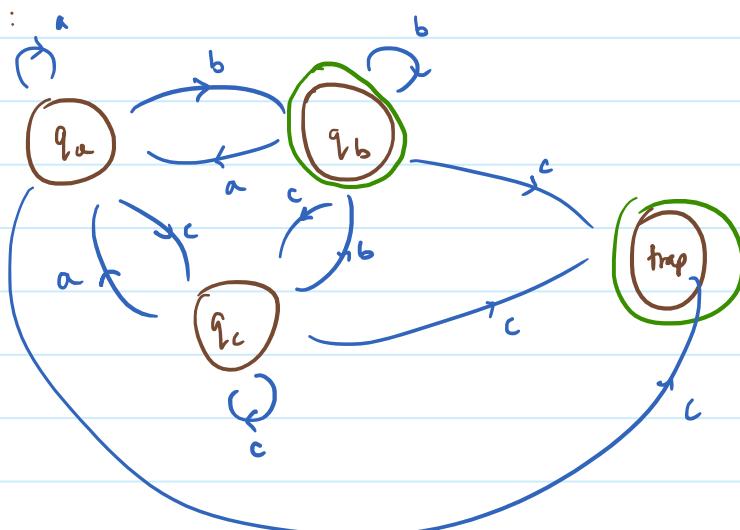
$$\Omega = \{(\emptyset, \{q_b\}), (\{q_a, q_b\}, \emptyset)\}$$

Muller - condition

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

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

Büchi - condition:



Thm (McNaughton)

DMA = DRA = DSA = NBA =  $\omega$ -regular.

D = deterministic

N = non-D

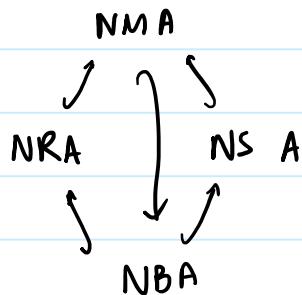
M = Muller, R = Rabin, S = Streetter, B = Büchi

A = Automaton

# Lecture 10 (03-09-2021)

01 September 2021 09:58

Theorem.



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  $A = (Q, \Sigma, q_0, \Delta, S_2)$  be a NRA.

$S_2 = \{(E_1, F_1), \dots, (E_k, F_k)\}$ .

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

- $NSA \hookrightarrow NMA$ : 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  $A_i := (Q, \Sigma, q_0, \Delta, \{x_i\})$ .

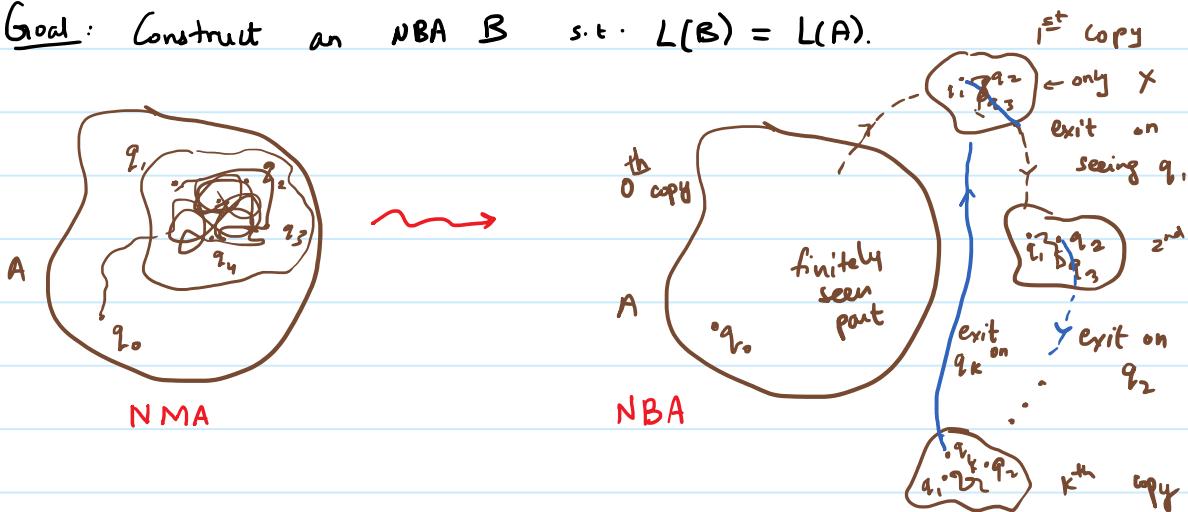
$$\text{Note: } L(A) = \bigcup_{i=1}^k L(A_i).$$

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

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

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

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



$B = (Q', (q_0, 0), \Sigma, \Delta', \{(q_1, 1)\}^2)$

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

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

$\stackrel{?}{\text{K+1=1}}$

the arrows here indicate those in original  $\Delta$

Thus, all these models have same expressive power.

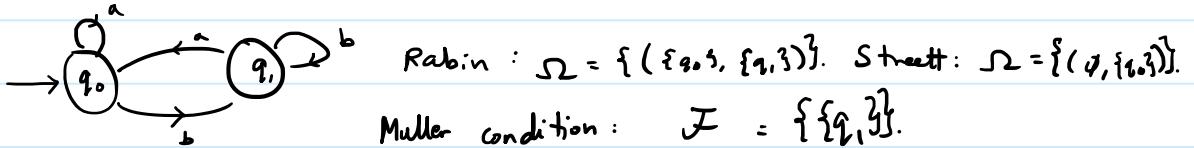
$$NRA = NFA = NSA = NBA$$

ut

DBA

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

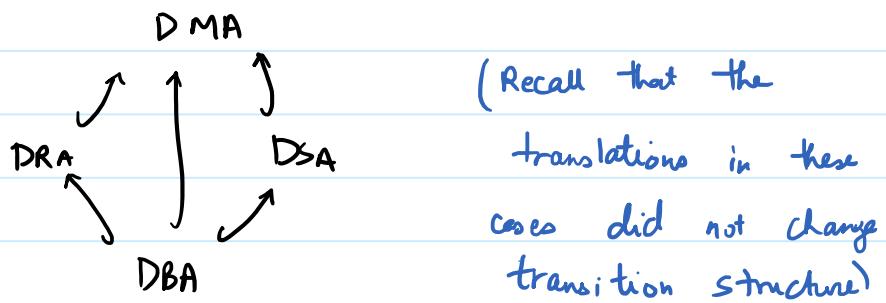
We had seen  $L = \{\alpha \mid \alpha \text{ has finitely many 'a's}\} \in \text{NSA} \setminus \text{DBA}$ .  
 (on  $\Sigma = \{a, b\}$ )



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

Then,  $L$  is accepted by this DMA.

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:

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

$$\parallel \quad \parallel \quad \parallel \quad \uparrow \text{#}$$

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

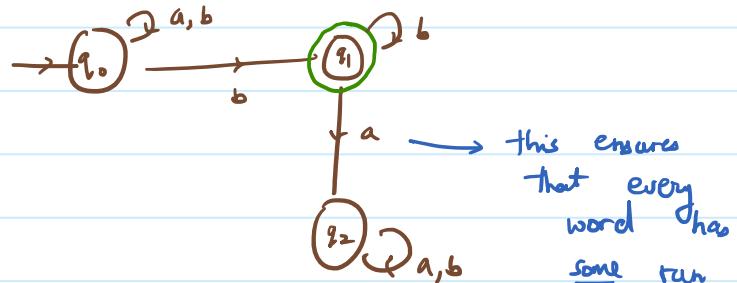
# Lecture 11 (03-09-2021)

03 September 2021 09:28

## Determinisation of Büchi automata:

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

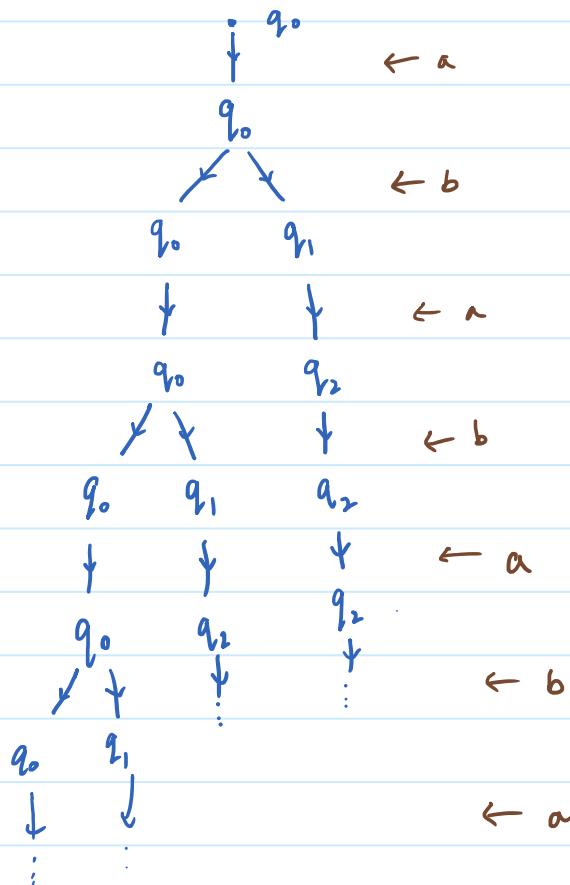
Finitely many 'a's :



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  $\alpha$ .

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



A run  $\gamma$  of  $A$  on  $\alpha$  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:

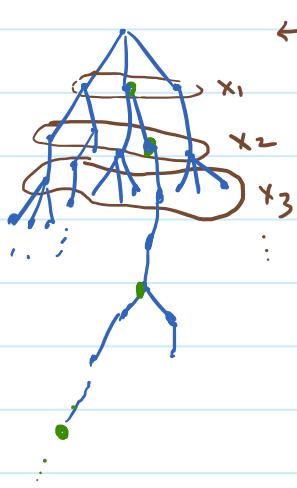
$$A_s = (2^Q, \Sigma, \{q_0\}, \delta_s, \text{acceptance}).$$

$\delta_s : 2^Q \times \Sigma \rightarrow 2^Q$  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  $\alpha$  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.

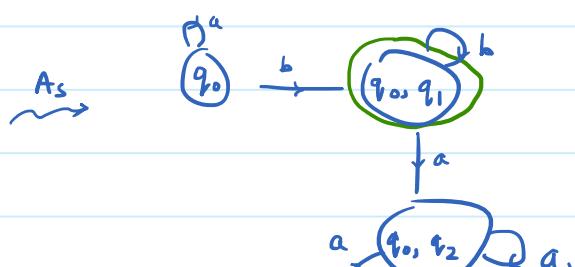
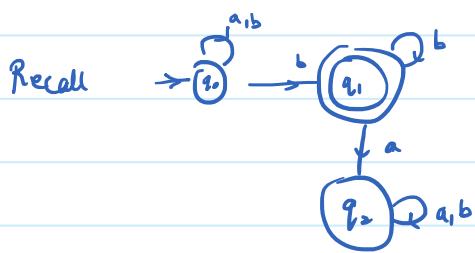
Define

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

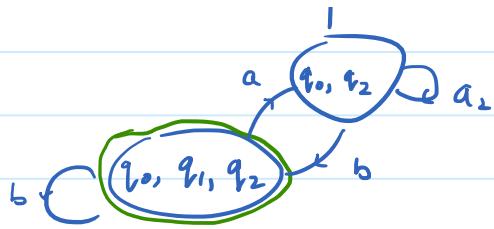
$\hookrightarrow$  good states

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  $\alpha$

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

is said to be strongly accepting if  $\exists i_0 < i_1 < i_2 < \dots$   
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  $\alpha$ .

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

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

( $\gamma \in \Gamma$ )

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

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

$$(x_1, x_1) \xrightarrow{u} (x_2, x_2) \Rightarrow \forall q_1 \in x_2 \exists q'_1 \in x_1 \text{ s.t.} \\ q'_1 \xrightarrow{u} q_1.$$

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

# Lecture 12 (08-09-2021)

08 September 2021 09:47

## Determinization

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

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

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

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

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

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

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

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

Recall proof.

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

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

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

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \xrightarrow{u} \begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

$y_i \downarrow$

$$\downarrow \begin{pmatrix} x_{i+1} \\ x_{i+1} \cap G \end{pmatrix} \xrightarrow{\quad} \dots$$

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

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

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

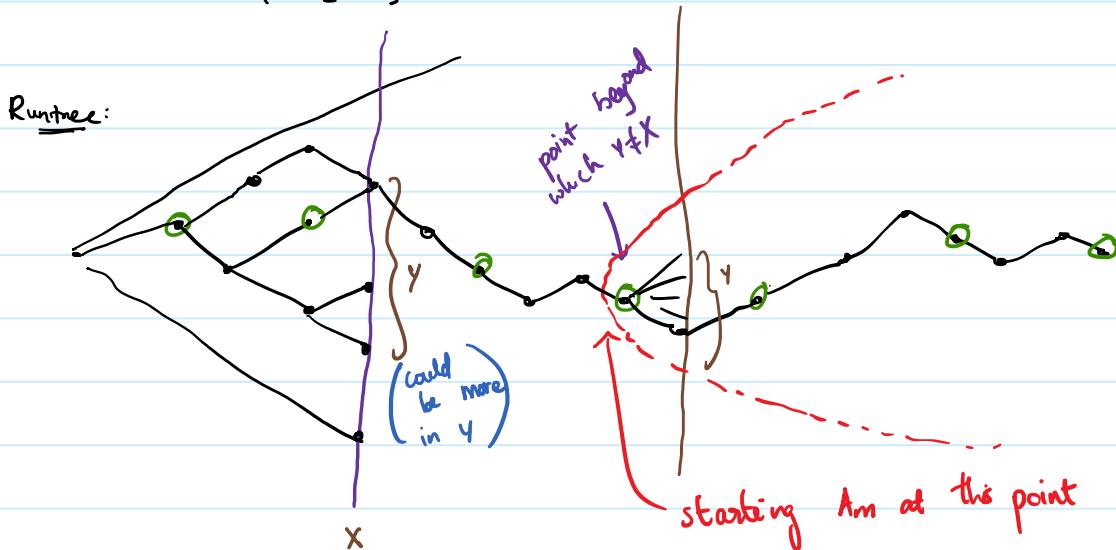
At this point, we applied König's lemma to see that A has an acc. run on  $\alpha$ . Thus,  $\alpha \in L(A)$ .



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

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

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



$\alpha$  not acc. by  $A_m \Rightarrow$  beyond a point,  $f_m$  does not encounter  $(x, x)$ .

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

## 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  $\alpha$  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  $\omega$ -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  $\omega$ -regular  $L$  can be written as

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

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  $q_0 \xrightarrow{u} \star g$   
and  $\beta$  is accepted by the  $A_m$  started at  $g$ .

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

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

$\downarrow$   
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  $U \cdot \overrightarrow{V}$ .

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

Thus, we conclude that every  $\omega$ -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

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

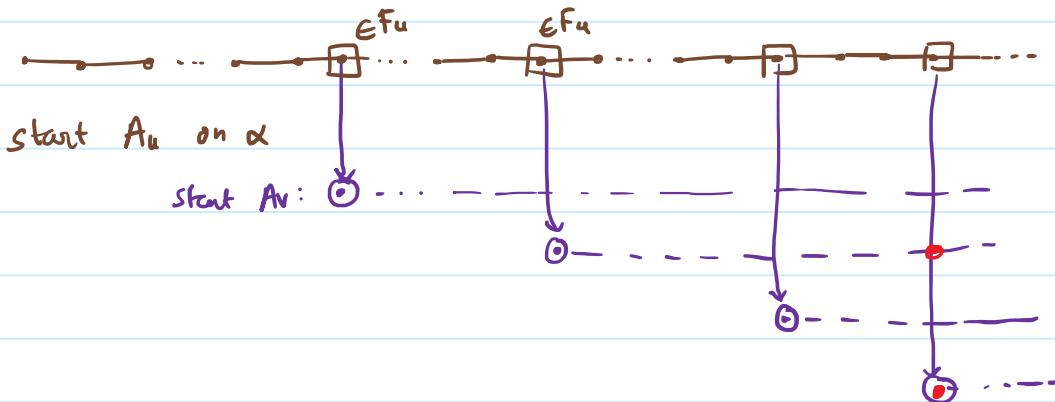
$$A_V = (Q_V, q_{\text{0}}^V, \Sigma, \delta_V, F_V).$$

Goal: To construct a deterministic  $\omega$ -automaton (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.

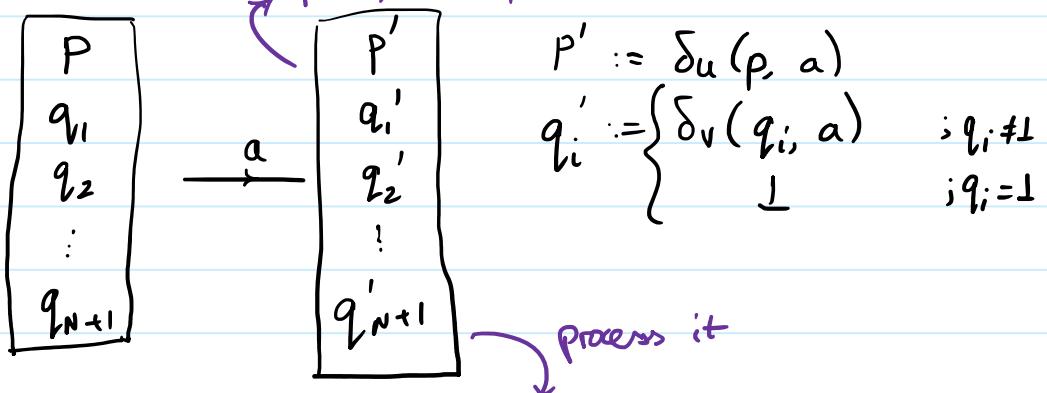
Initial state:

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

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

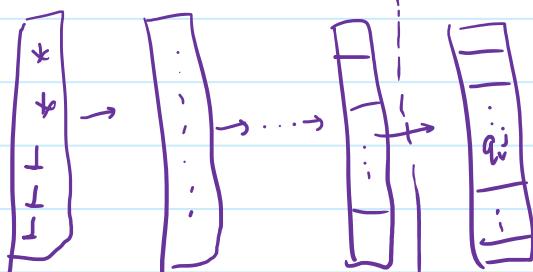
After this processing, whatever we are left with, is the new state.

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

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

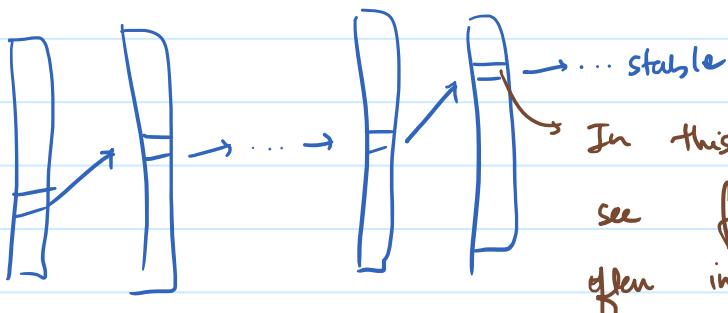
$\Rightarrow - (\omega_1, \omega_2, \dots, \omega_i, \dots)$  on  $\omega$ .

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



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  $\gamma$  of  $B$  on  $\alpha$  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  $\rho$  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  $\rho$  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  $\omega$ -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  $\omega$ -regular language is a boolean combination of languages accepted by DBA (or equivalently, languages of the form  $\overrightarrow{U}$ ).  $\nearrow \wedge, \vee, \neg$

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  $L_q = \{ \alpha : \text{unique run of } A \text{ on } \alpha \text{ visits } q \text{ inf. often} \}$   
 $= \vec{U}_q$ , 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 :  $A = (Q, q_0, \Sigma, \delta, \epsilon)$ .

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

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

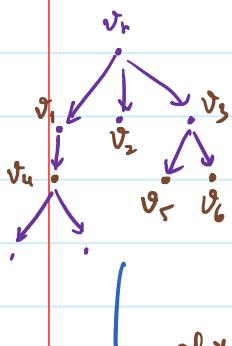
Notation. A tree  $T = (V, \mathcal{V}, \pi)$  where

$V \rightarrow$  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  $\text{children}(v)$ .

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

$$\text{Notation} \quad n := |\mathcal{A}|, \quad \mathcal{L} := \{\ell_1, \dots, \ell_{2n}\}.$$

$\hookrightarrow$  set of labels

Def'n. A Suffix tree is  $s = (T, \sigma, x, \gamma)$ , where

- $T = (V, \vartheta_r, \pi)$  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  
 $\sigma(v) \cap \sigma(v') = \emptyset$ .

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

- $\lambda : V \hookrightarrow L$  is injective.

Different vertices are assigned different labels.

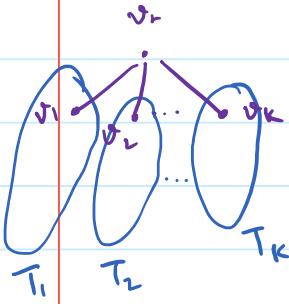
Note: The above implicitly forces  $|V| \leq 2n$ . In fact, now is true.

Claim. Let  $S = (T, \sigma, \chi, \lambda)$  be a Safr tree with  $T = (V, \sigma_r, \pi)$ . 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  
 $\sigma(v_i) \subsetneq \sigma(v_r)$

$\sqcup \sigma(v_i) \subsetneq \sigma(v_r)$

□

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

---

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

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  $v$ .

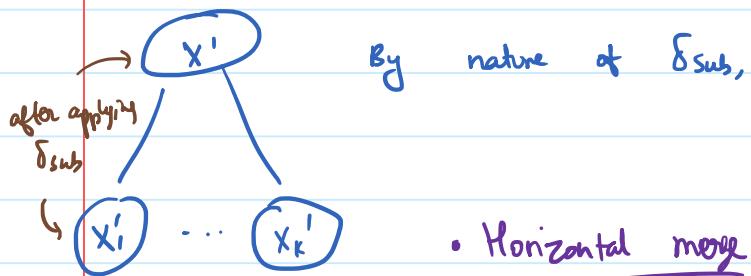
- Define  $\sigma'_i$  of such a  $v'$  as  
 $\sigma'_i(v') = \sigma(v) \cap G_i$ .

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

- 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  $\delta_{\text{sub}}$ , we have  $x'_i \subseteq x'$   
 $\forall i$ .

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

- Vertical merge

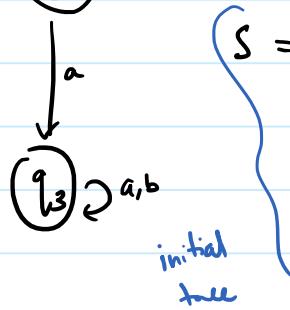
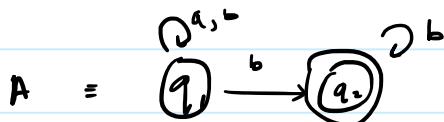
If  $v = \bigcup_{w \in \text{children}(v)} \sigma(w)$ , delete all children  
of  $v$ .

. 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.



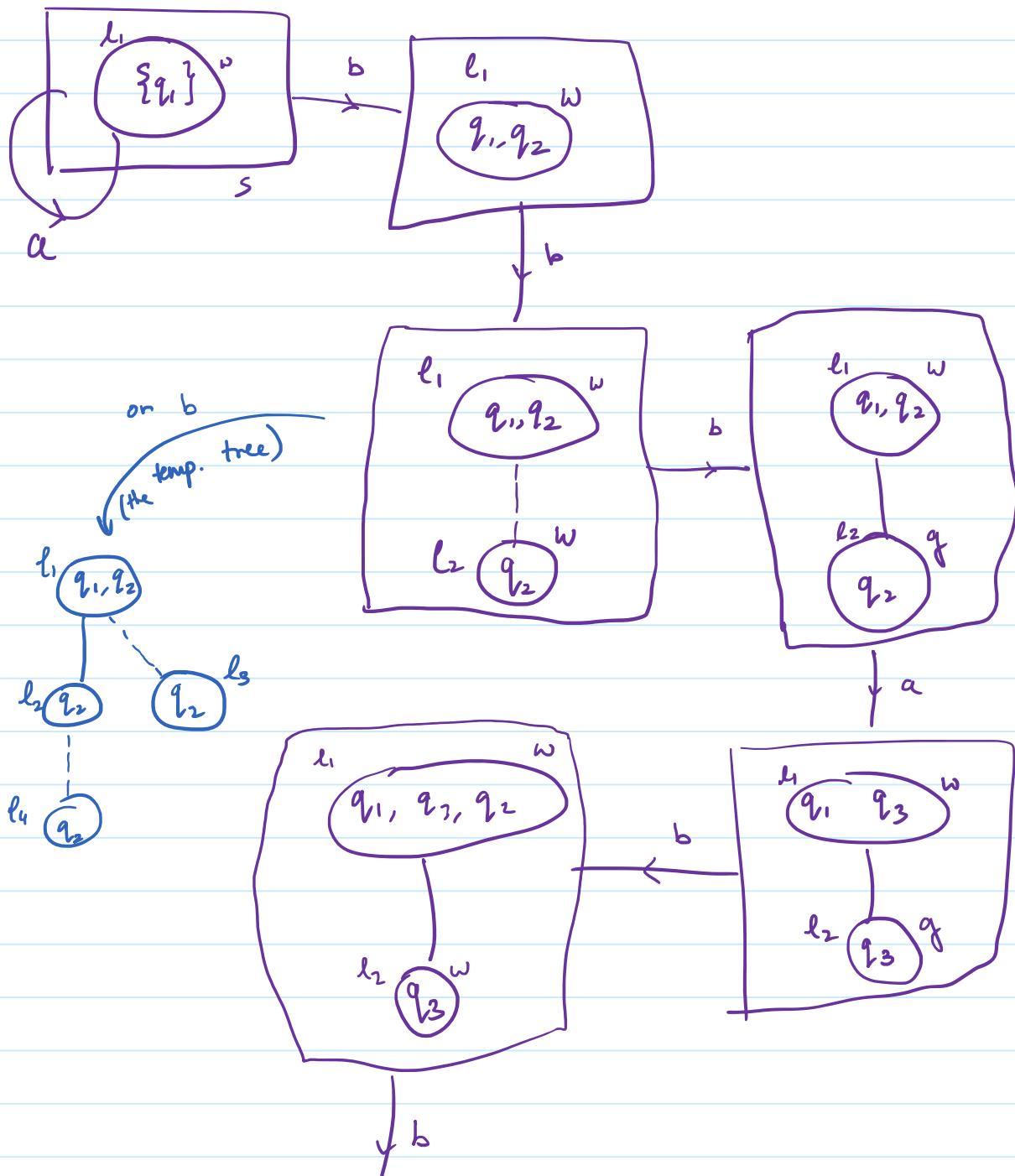
$$\mathcal{L} = \{l_1, \dots, l_6\}$$

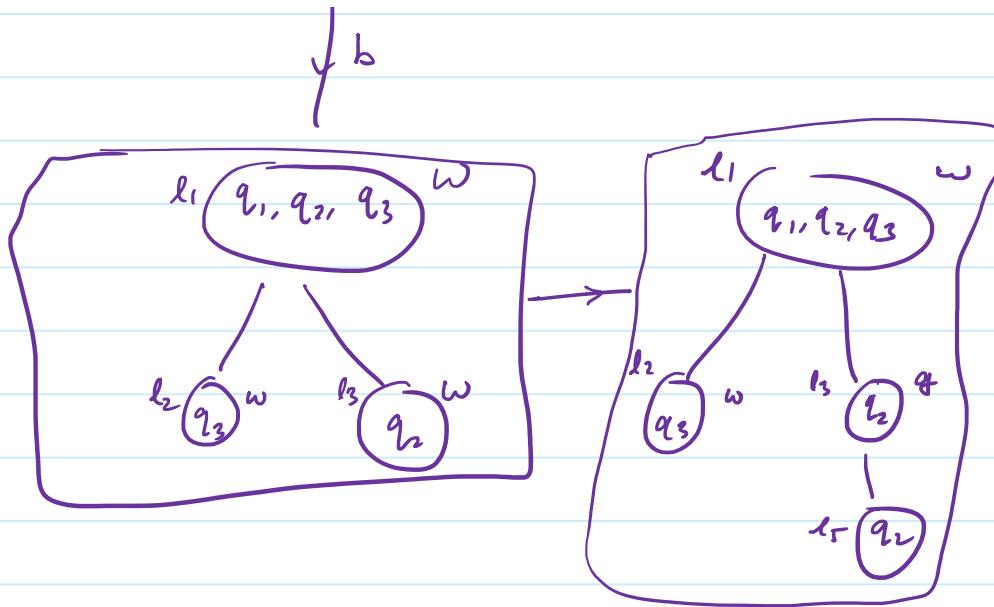
$$S = (T, \sigma, x, \lambda), \quad T = (V, V_r, \pi)$$

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

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

$$\lambda : V \rightarrow \mathcal{L}$$





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

Büchi Automaton

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



$$\mathcal{R} = \{(E_1, F_1), \dots, (E_n, F_n)\}.$$

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

$F_i = \{s \in S : s \text{ has a node } v \text{ s.t.}$

$$\gamma(v) = l_i \text{ and } \chi(v) = \text{green}\}$$
.

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

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

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

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)  $x_j(v) = x_k(v) = \text{green}$   
and  $\forall i \in \{j+1, \dots, k-1\}: x_i(v) = \text{white}.$