§1 05-11

Σ corresponds to the alphabet. Σ^* represents finite words. Σ^ω is the set of infinite words over Σ.

If $\Sigma = \{a, b\}$, Σ^ω is uncountable. $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. A string in Σ^ω is countable, so positions in the string can be indexed by $\mathbb{N}$ i.e. countable.

There is a natural partial order on Σ^∞. $\alpha \leq \beta$ if α is a prefix of β. Infinite strings cannot be compared.
First ordinal. Reflexivity is part of a partial order.

Remark 1.1. We can define a metric on Σ^* by letting $d(x, y) = 2^{-n}$ where n is the first position where $x[n]! = y[n]$. The longer the shared prefix the closer two words are. This is an ultrametric which means that it satisfies the following strengthened version of the triangle inequality

$$d(x, z) \leq \max(d(x, y), d(y, z))$$

Σ^∞ is the cauchy completion of Σ^* and the metric naturally extends to Σ^*.

Infinite strings are important for the verification of reactive systems. They are designed to run forever so they have to talk about infinite computations.

§1.1 ω regular expressions.

Let E be a regular expression and $\epsilon \notin L(E)$.
E^ω is an ω-regular expression.

$$L_\omega(E^\omega) := \{w_1 w_2 w_3 \cdots \mid w_i \in L(E)\}$$

where $L_\omega(F)$ is the ω-language defined by the ω regular expression F.
Let E_i be regular xpressions and F_i be regular expressions such that $\epsilon \notin L(F_i)$.

$$E_1(F_1)^\omega + \cdots + E_n(F_n)^\omega$$

is a general ω-regular expression.
$E^\omega = EE^\omega$ so E^ω is also an ω-regular expression.

When concatenating where $\alpha, \beta \in \Sigma^\omega$, $\alpha\beta = \alpha$.

Definition 1.2. An ω-language ($\subseteq \Sigma^\omega$) is ω-regluar if defined by some ω regular expression. This ω-regular expression may not be unique.

Note 1.3. $\cdot$ is concatenation. $+$ is union. $(.)^\omega$ is infinite repetition.

§1.2 ω-automata

Specifically non deterministic biichi automata.

$$A = (Q, \Sigma, \delta, Q_0, F)$$
$$\delta : Q \times \Sigma \to 2^Q$$
$$s' \in \delta(s, a) \equiv s \to_a s'$$

Given a word $\sigma \in \Sigma^\omega$, a run of ω in A is an infinite sequence of states $q_0, q_1, \cdots$ such that $q_0 \in Q_0$, $\forall i\ q_i \rightarrow_{\sigma[i]} q_{i+1}$.

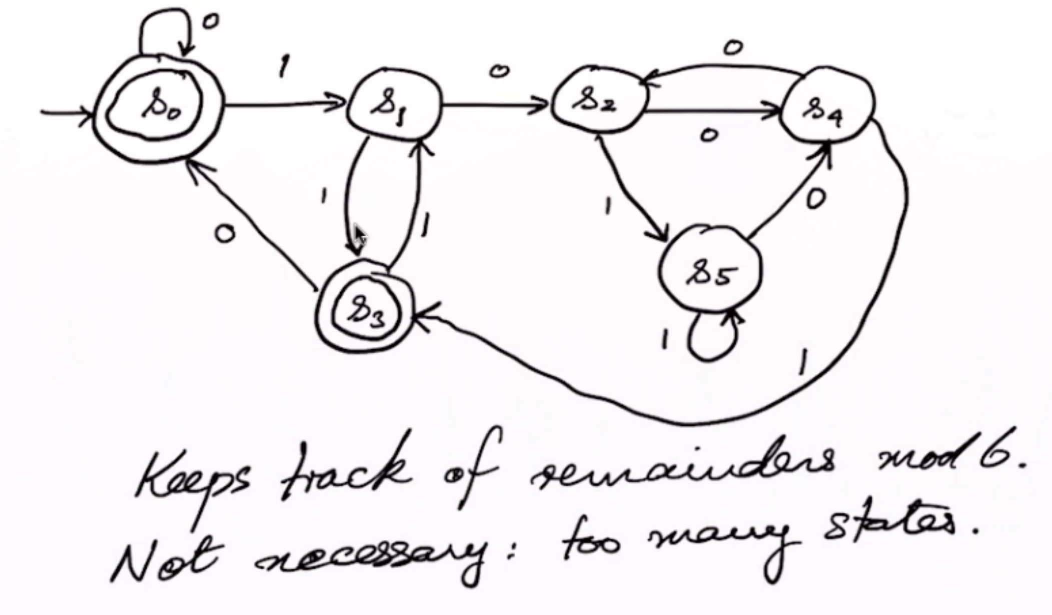
$$inf(r) = \{q \mid q \text{ occurs infinitely often in} r\}$$

A run is accepting if it hits an accept state infinitely often. There are only finitely many accept states so one of them has to be hit infinitely many times.

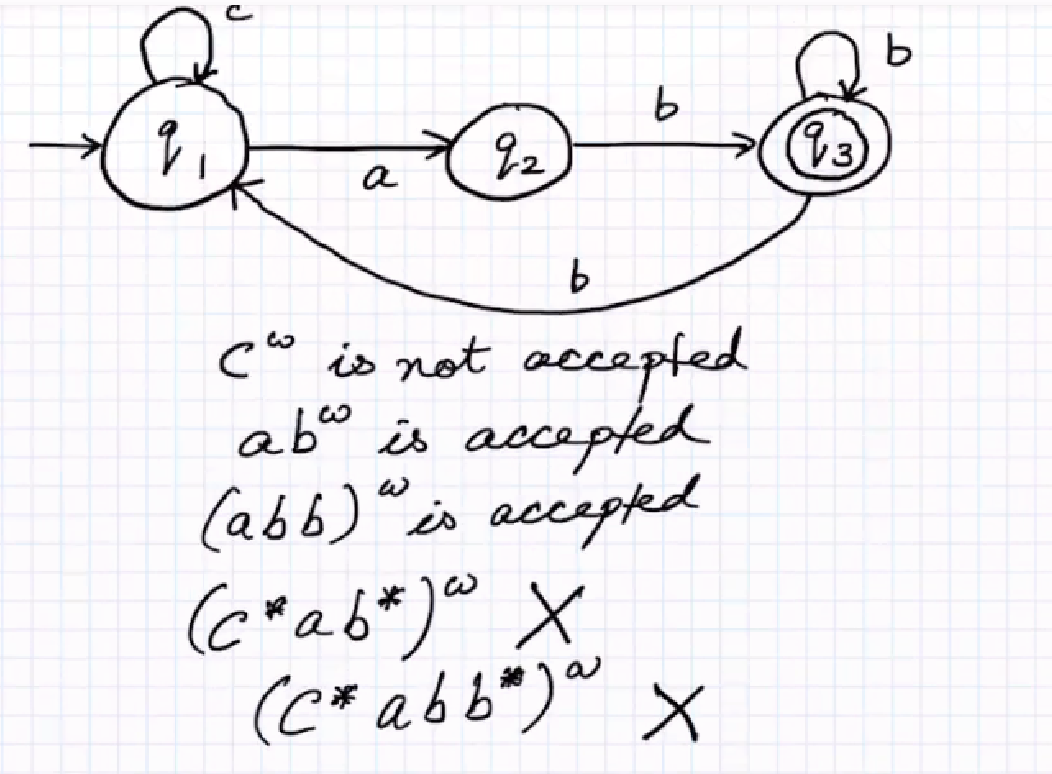
$$inf(r) \cap F \neq \varnothing$$

A word is accepted by A if that word has some accepting run.

Example 1.4



Keeps track of remainders mod 6.
Not necessary: too many states.



c^ω is not accepted
ab^ω is accepted
$(abb)^\omega$ is accepted
$(c^*ab^*)^\omega$ X
$(c^*abb^*)^\omega$ X

$$L_\omega(A) = c^*ab(b^+ + bc^*ab)^\omega$$

We see that it doesn't necessarily have to be periodic, but perhaps a periodic aspect to it. Get it to the start state first, then think of a pattern.

Theorem 1.5

NBA's accept exactly the ω-regular languages.

Proof. Show that any ω-regular language can be captured by an NBA.

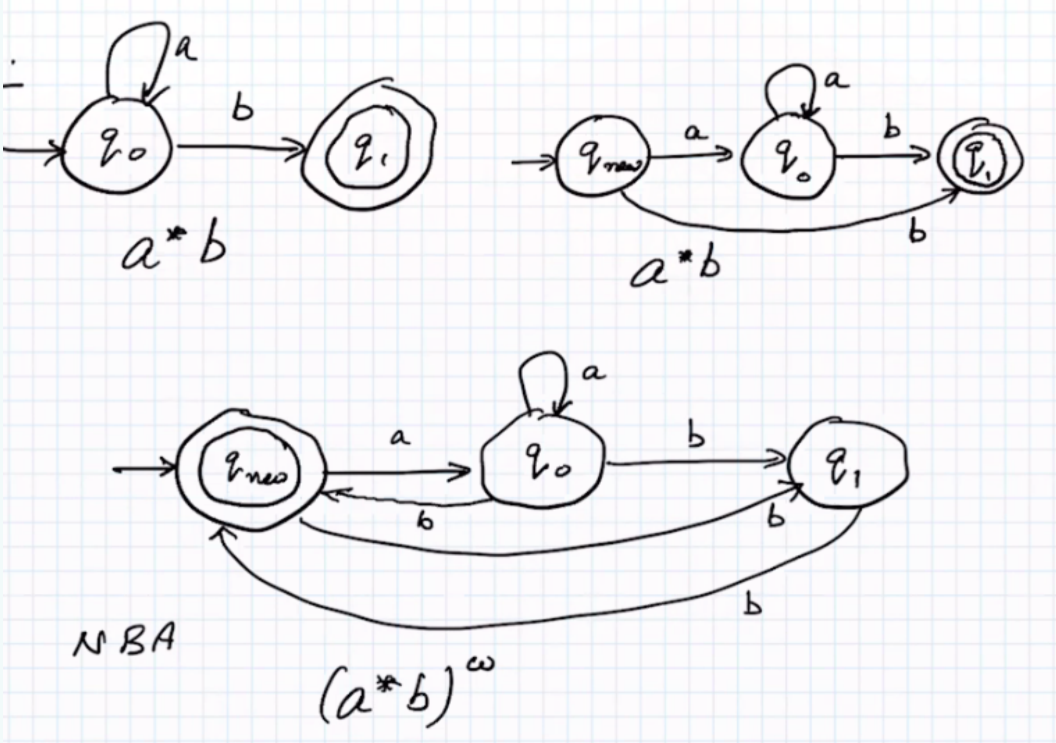
Given $A_i = (Q_i, \Sigma, \delta_i, Q_{0,i}, F_i)$, $i = 1, 2$ with $Q_1 \cap Q_2 = \varnothing$. Then $A_1 + A_2 = (Q_1 \cup Q_2, \Sigma, \delta, Q_{0,1} \cup Q_{0,2}, F_1 \cup F_2)$. You can start either in start states of A_1 or A_2.

$$\delta(q, a) = \delta_i(q, a) \text{ if } q \in Q_i$$
$$L_\omega(A_1 + A_2) = L_\omega(A_1) \cup L_\omega(A_2)$$

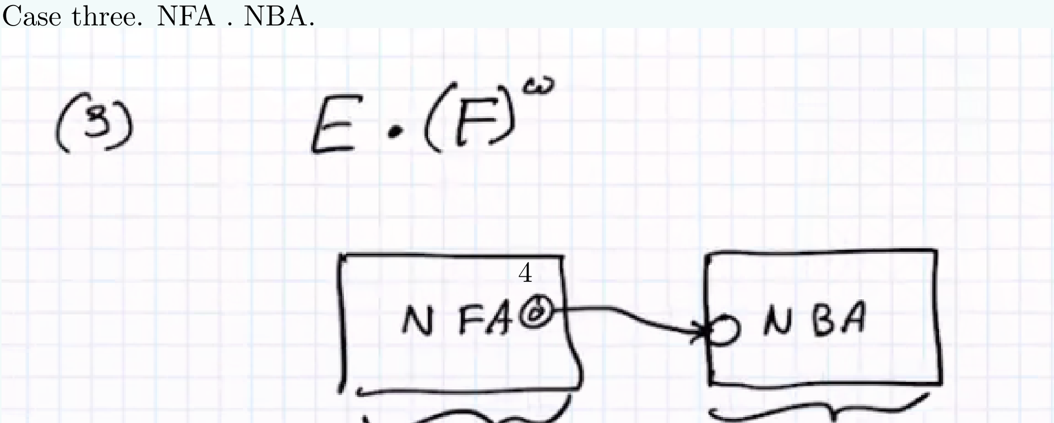
We are now considering F^ω. We have an NFA for F and assume that all the inital states are non accepting. $Q_0 \cap F = \varnothing$ and there are no incoming transition to any state in Q_0. You may worry that this will be too restricting on applicable NFAs. But if your NFA does not satisfy these assumptions we can modify it so that it does and defines the same language.

Idea for doing this: Brand new unique start state. And then leave everything afterwards untouched.

Given $A = (Q, \Sigma, \delta, Q_0, F)$. NBA $A' = (Q, \Sigma, \delta', Q_1, F')$. $L_\omega(A') = (L(A))^\omega$. The goal is to keep recognizing infinitely many words. So each time you hit an accept state, send it back to the start state.



Case three. NFA . NBA.



Theorem 1.6

Other direction

Suppose $A = (Q, \Sigma, \delta, Q_0, F)$ NBA. Fix any 2 states. $A_{qp} := (Q, \Sigma, \delta, \{q\}, \{p\})$. This converts it to an NFA. It is not meant to read omega words.

$$L(A_{qp}) = L_{qp} = \{w \in \Sigma^* \ |\delta^*(q, w) \ni p\}$$

These are words that take you from q to p.

Now consider an accepted word σ of $L_\omega(A)$ with an accepting run. It must hit sum $q \in F$ that appears infinitely often. Therefore

$$\sigma = \underbrace{w_0}_{L_{q_0 q}} \underbrace{w_1}_{L_{qq}} \underbrace{w_1}_{L_{qq}} \cdots$$

where w_i are non empty. Any σ the can be expressed like this has an accepting run. Therefore

$$L_\omega(A) = \cup_{q_0 \in Q_0} \cup_{q \in F} L_{q_0 q} \cdot (L_{qq} \setminus \{\epsilon\})^\omega$$

Therefore anything recognized by an NBA is described by an ω-regular expression. Kleene's theorem for ω-languages.

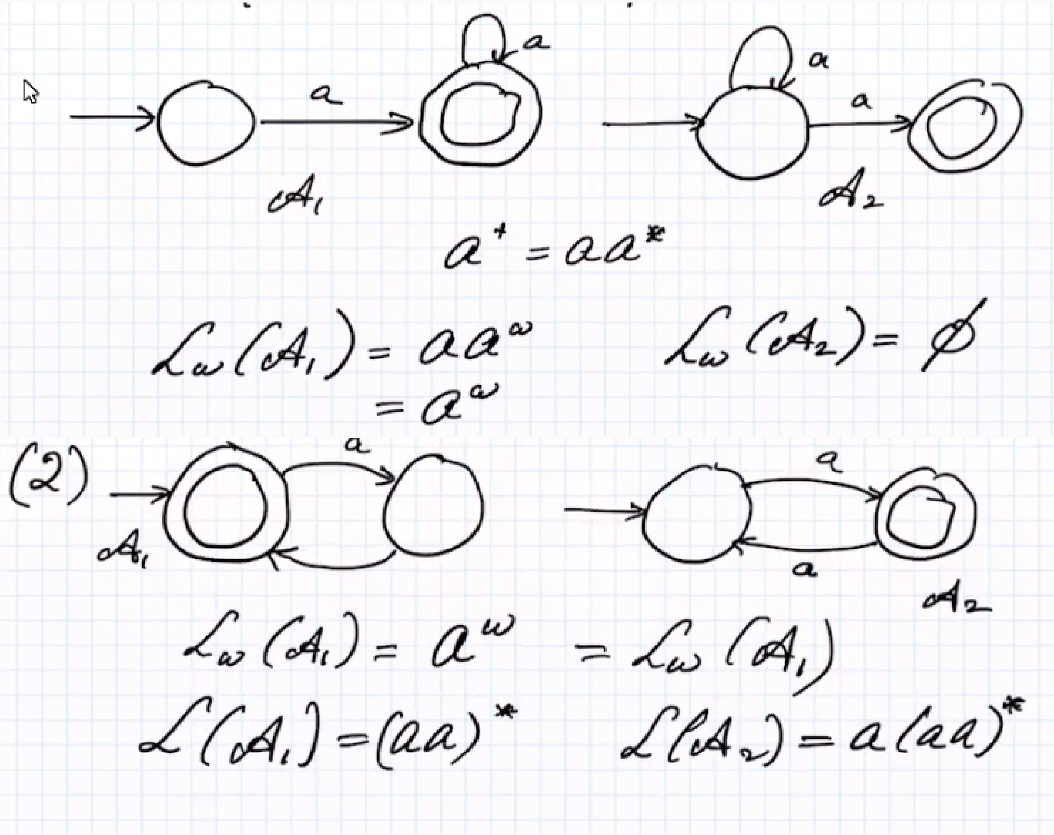
Lemma 1.7

$L_\omega(A) \neq \varnothing$ iff $\exists$ a reachable accept state that belongs to a cycle.

$A_1 \equiv_F A_2$ they are the same as NFA. $L(A_1) = L(A_2)$. $A_1 \equiv_B A_2$ they are the same as NBA. $L_\omega(A_1) = L_\omega(A_2)$.

Example 1.8

Does $\equiv_F \Rightarrow \equiv_B$? No! Does $\equiv_B \Rightarrow \equiv_F$? No!



$$a^+ = aa^*$$

$$\mathcal{L}_\omega(\mathcal{A}_1) = aa^\omega = a^\omega \qquad \mathcal{L}_\omega(\mathcal{A}_2) = \emptyset$$

$$\mathcal{L}_\omega(\mathcal{A}_1) = a^\omega = \mathcal{L}_\omega(\mathcal{A}_1)$$

$$\mathcal{L}(\mathcal{A}_1) = (aa)^* \qquad \mathcal{L}(\mathcal{A}_2) = a(aa)^*$$

Idea here is that it distinguishes even length from odd length, but infinity length is both even and odd.

Theorem 1.9

NBAs and DBAs are not equivalent. There is no DBA that can recognize $(a+b)^*b^\omega$. This is what makes omega automata much trickier.

Proof. Suppose we have a DBA for this language. We can define δ^* as usual on the underlying DFA. Consider the word $b^\omega \in (a+b)^*b^\omega$

Since b^ω is accepted, we must hit an accept state at some point. Therefore $\delta^*(q_0, b^{n_1})$ is an accept state. $b^{n_1}ab^\omega \in L$. This has to hit an accept state again after reading a. You can continue this process. Have to hit the same accept state twice. But then $\exists i, j, i < j$ where

$$\delta^*(q_0, b^{n_1}ab^{n_2}\cdots ab^{n_i}) = \delta^*(q_0, b^{n_1}ab^{n_2}\cdots ab^{n_j})$$

But then there is a loop. So you can include infinitely many b in the language. Contradiction. So NBA are strictly more expressive than DBA. □

Let $\alpha \in \Sigma^*\omega$ and define $\alpha|_n \in \Sigma^*$ to be the length of n finite prefix.

Definition 1.10. For $W \subseteq \Sigma^*$ we define $\vec{W} = \{\alpha \in \Sigma^\omega \mid \exists$ infinitely many $n \in \mathbb{N} s.t. \alpha|_n \in W\}$ Words where you can always make the prefix in W longer. $\vec{W}$ is called a <u>limit</u> language.

Theorem 1.11

An ω-regular language L is recognizable by a deterministic BA iff there is a regular language W such that $L = \vec{W}$.

Proof. DIY □

Theorem 1.12

DBA are closed under complement.

If A is a DBA, $\exists$ an NBA A' such that $L_\omega(A') = \Sigma^*\omega \setminus L(A)$.

$A' = (Q', \Sigma, \delta', Q'_0, F').A = (Q, \Sigma, \delta, Q_0, F)$.

$$Q' = (Q \times \{0\}) \cup ((Q \setminus F) \cup \{1\})$$
$$Q'_0 = Q_0 \times \{0\}$$