

To-do:

- HW1 - Due \sim 1 wk before Quiz.
↳ Sections 1.1, 1.2

Hw: 20% (4 ass) — no late ass

Q1: 10% - in class Jan 29

Midterm: 25% - Fri 18:30-20:00 Mar 1

Q2: 15% Mar 14 (in class)

Final: 30%

[CS, Waterloo.ca/~watrous/c53](http://cs Waterloo.ca/~watrous/c53)

John Watrous

Qn C 3312

OH: Tues 1-3 pm or by appointment

Theory of Computing:

mathematics of abstract models of computation, and the problems they can (or cannot) solve.

Foundations & Defs:

- Assume comfortable with set theory
- finite Set S has size $|S|$
- Sets can be infinite

i.e. $\mathbb{N} = \{0, 1, 2, \dots\}$ (^{natural}_{numbers})

$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ (^{integers})

$\mathbb{Q} = \{x: x = \frac{n}{m}, n, m \in \mathbb{Z}, m \neq 0\}$ (^{rational #s})

\mathbb{R}, \mathbb{C} real & complex #s.

These are inf sets, but we need a more refined notion

Def: A set A is countable if there exists any one-to-one (injective) function f of the form:

$$f: A \rightarrow \mathbb{N}$$

These are equivalent (for any set A)

1. A is countable
2. either A is empty, or there is an onto (or surjective) function

$$g: \mathbb{N} \rightarrow A$$

- 3 Either A is finite or there exists a one-to-one and onto (or bijective) function.

$$h: \mathbb{N} \rightarrow A$$

Ex.

\mathbb{N} is countable

\mathbb{Z} is countable

$f: \mathbb{Z} \rightarrow \mathbb{N}$ def as

$$f(0) = 0, f(n) = 2n,$$

$$f(n) = 2n+1$$

for all $n \geq 1$

\mathbb{Q} is countable

0

list 0

-1, +1

1: s+1

-2, $-\frac{1}{2}$, $\frac{1}{2}$, 2

list 2

-3, $-\frac{3}{2}$, $-\frac{2}{3}$, $-\frac{1}{3}$, $\frac{1}{3}$, $\frac{2}{3}$, $\frac{3}{2}$, 3

list 3

:

{...} list k

Sorted list of all fractions

$$\pm \frac{n}{m} \quad 1 \leq n, m \leq k,$$

either $n=k$ or $m=k$
and $\gcd(m, n)=1$

Is every set countable? No.

For any set A , the Power set of A is the set of all subsets of A .

Notation: $P(A)$ $A = \{1, 2, 3\}$
 $P(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

The set $P(\mathbb{N})$ is not countable

Pf:

- Assume it is countable (toward contradiction)
- This implies there exists an Onto function.

$$g: \mathbb{N} \rightarrow P(\mathbb{N})$$

Define $S = \{n \in \mathbb{N} : n \notin g(n)\}$

g is onto, and $S \in P(\mathbb{N})$ so there must exist at least one number $n \in \mathbb{N}$ such that $S = g(n)$.

Is $n \in S$?



Is $n \in S$?

$n \in S \Leftrightarrow n \in g(n)$ (because $S = g(S)$)

$n \in S \Leftrightarrow n \notin g(n)$ (definition of S)

So

$n \in g(n) \Leftrightarrow n \notin g(n)$

Contradiction.

Comments:

- Proof technique is called diagonalization



Terminology:

Alphabet: Any finite, non-empty set. we view these as symbols

A string over an alphabet Σ is a finite, unordered sequence of symbols from $\Sigma \rightarrow$

01101 - String

01101... - not a string

ϵ - empty string

length of a string : it is the total number of symbols.

01101 → length = 5

$\epsilon \rightarrow \text{length} = 0$

|w| is the length of string w.

$$|01101| = 5$$

A language over an alphabet Σ is a set of strings over Σ

$L = \{w : w \text{ is a string over } \{0,1\}, |w| \text{ is even}\}$

For any alphabet Σ , the language containing all strings over Σ is countable

$\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots$

0 1 2 3 4 5 6 7 8 ...

N

A set A is countable iff; it is the empty set, or there is an onto function $f: \mathbb{N} \rightarrow A$

Σ = alphabet

Σ^* = set of all strings over Σ

↳ For $\Sigma = \{0, 1\}$ Σ^* is countable.

$\epsilon, 0, 1, 00, 01, 10, 11, \dots$

Proposition: Σ^* alphabet
 $\Rightarrow P(\Sigma^*)$ is not countable

Cantor: $P(\mathbb{N})$ is uncountable.

Have a bijection $f: \mathbb{N} \rightarrow \Sigma^*$
Define $g: P(\mathbb{N}) \rightarrow P(\Sigma^*)$

$$g(A) \stackrel{\text{def}}{=} \{f(n) : n \in A\}$$

Since $P(\mathbb{N})$ is uncountable and there
is a bijection $P(\mathbb{N}) \rightarrow P(\Sigma^*)$,
then $P(\Sigma^*)$ is uncountable as well

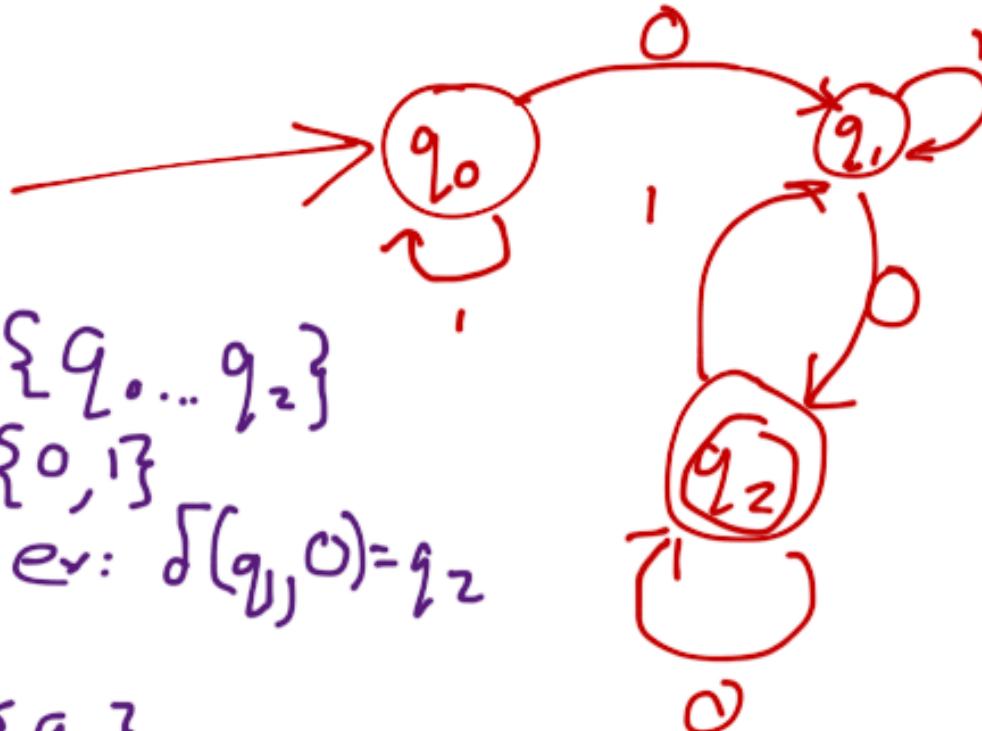
DFA_s

A DFA is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

where

- Q is a finite, non-empty set whose elements we call states
- Σ is an alphabet
- δ is a function (a transition function) of the form: $\delta: Q \times \Sigma \rightarrow Q$
- q_0 is the start state.
- $F, F \subseteq Q$, is a subset of states - "accept states"



$$Q = \{q_0, \dots, q_2\}$$

$$\Sigma = \{0, 1\}$$

$$\delta = \dots \text{ ex: } \delta(q_0, 0) = q_2$$

$$q_0 = \dots$$

$$F = \{q_2\}$$

Definition:

Supp: $M = (Q, \Sigma, \delta, q_0, F)$
 is a DFA and $w \in \Sigma^*$. The DFA M
accepts w iff

1. $w = \epsilon$, and $q_0 \in F$, or
2. $w = q_0 \dots q_n$ for $n \geq 1$, $q_1 \dots q_n \in \Sigma$ and
 there exist states $r_0 \dots r_n \in Q$
 such that $r_0 = q_0$, $r_n \in F$ and
 $\delta(r_k, q_{k+1}) = r_{k+1}$ for all $k \in \{0 \dots n-1\}$

An equivalent def:

- extend δ to operation $r \in \Sigma^*$.

Define:

$$\delta^*: Q \times \Sigma^* \rightarrow Q$$

as follows:

1. $\delta^*(q, \epsilon) = q$
2. $\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$ for all $q \in Q$, $w \in \Sigma^*$,
 $\sigma \in \Sigma$

M accepts w iff $\delta^*(q_0, w) \in F$

If M does not accept w , we say it rejects w .

The lang recognized by M is:

$$L(M) = \{w \in \Sigma^*: M \text{ accepts } w\}$$

Definition:

A language $A \subseteq \Sigma^*$ over any alphabet Σ is regular iff there exists a DFA M with $A \equiv L(M)$

How many regular languages are there
(over a given Σ)? \rightarrow Countable.

$n \cdot n \cdot 2^{|\Sigma|}$
↑.↓.asim & accept

NFA_s

A NFA is a 5-tuple

$$\underline{N} = (Q, \Sigma, \delta, q_0, F)$$

where

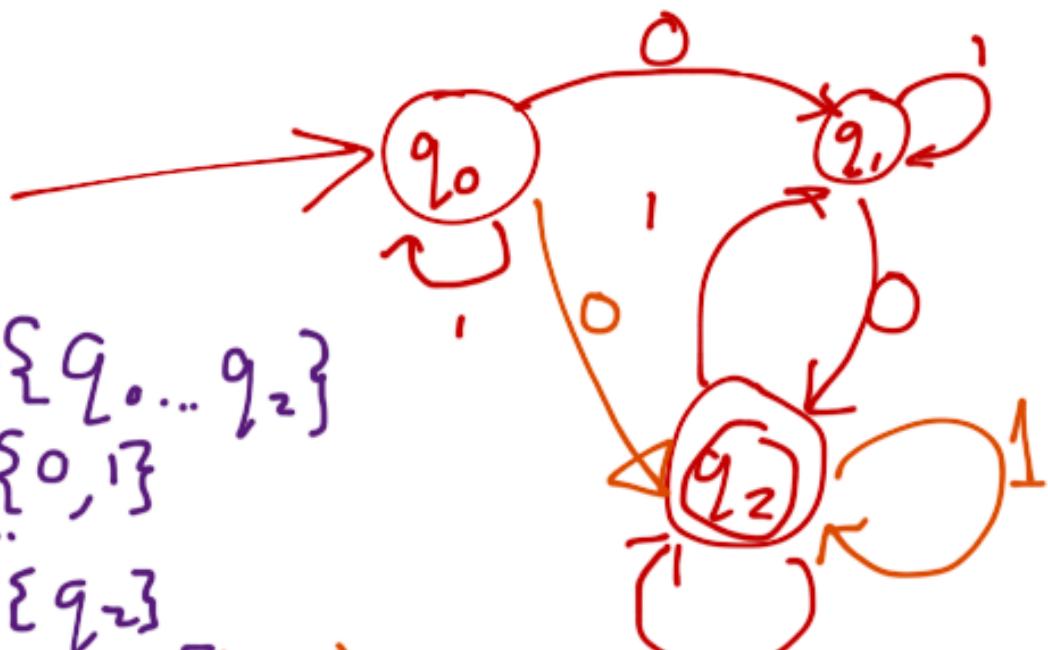
→ Q is a finite, non-empty set whose elements we call states

→ Σ is an alphabet

→ q_0 is the start state.

→ $F, F \subseteq Q$ is a subset of states - "accept"

→ $\delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$.



$$Q = \{q_0, \dots, q_2\}$$

$$\Sigma = \{0, 1\}$$

$$q_0 = \dots$$

$$F = \{q_2\}$$

$$\delta = \dots \text{ ex: } \delta(q_0, 0) = \{q_1, q_2\}$$

Definition:
Supp: $N = (Q, \Sigma, \delta, q_0, F)$
is an NFA and $w \in \Sigma^*$. The NFA N
accepts w iff there exist:

1. $m \in \mathbb{N}$
2. $r_0, \dots, r_m \in Q$, and
3. $\sigma_1, \sigma_2, \dots, \sigma_m \in \Sigma \cup \{\epsilon\}$

such that

$r_0 = q_0$, $r_m \in F$, $w = \sigma_1 \dots \sigma_m$, and

for all $k \in \{0, \dots, m-1\}$ $r_{k+1} \in \delta(r_k, \sigma_{k+1})$

Alternatively, $\delta^*: Q \times \Sigma^* \rightarrow P(Q)$ as follows:

For any $R \subseteq Q$, define

$$E(R) = \left\{ q \in Q : q \text{ can be reached from one } r \in R \text{ by following } \geq 0 \text{ } \epsilon \text{ transitions} \right\}$$

$$\begin{aligned}\delta^*(q, \epsilon) &= E(\{q\}) \\ \delta^*(q, \omega\sigma) &= E\left(\bigcup_{r \in \delta^*(q, \omega)} \delta(r, \sigma)\right)\end{aligned}$$

now, N accepts ω iff

$$\delta^*(q_0, \omega) \cap F \neq \emptyset$$

$$L(N) = \{\omega \in \Sigma^* : N \text{ accepts } \omega\}$$

Thm: Let Σ be an alphabet and let
 $A \subseteq \Sigma^*$ be a language.
A is regular iff $A = L(N)$ for NFA N .
i.e. $NFA \leftrightarrow DFA$ conversion

\rightarrow Given NFA $N = (Q, \Sigma, \delta, q_0, F)$
want DFA $M = (P, \Sigma, \gamma, p_0, G)$
so that $L(N) = L(M)$

Step 1: construction

Step 2: prove it works.

Continuing...

Supp. $N = (Q, \Sigma, \delta, q_0, F)$ is an NFA. It follows that $L(N)$ is regular.

Want: DFA $M = (P, \Sigma, \gamma, p_0, F)$ such that $L(M) = L(N)$

If it has an NFA or DFA, it is regular

1. construct M .

2. prove $L(M) = L(N)$

1. Construct M

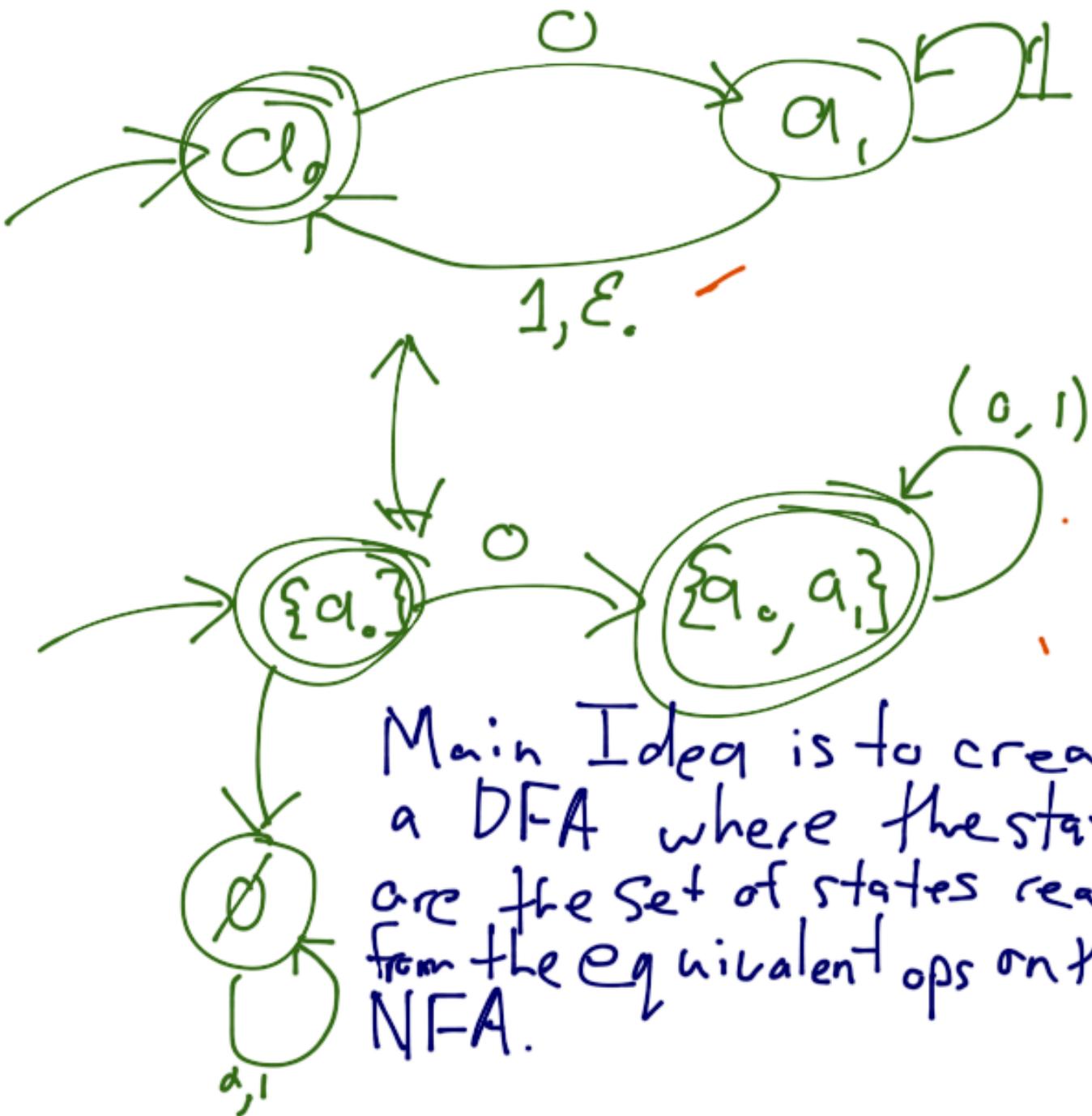
• $P = P(Q)$ (power set) $\{P \subseteq Q, E(P) = P\}$

• $p_0 = E(q_0)$ (epsilon closure)

• $G = \{P \in P : P \cap F \neq \emptyset\}$

(all possible states that could have been after this sequence)

• $\gamma(p, \sigma) = E \left[\bigcup_{q \in p} \delta(q, \sigma) \right]$



Main Idea is to create a DFA where the states are the set of states reachable from the equivalent ops on the NFA.

$\cap^*(E(p), \omega) = \bigcup_{q \in P} \delta^*(q, \omega)$ } if we prove
 for all $p \in P, \omega \in \Sigma^*$ } this, we prove
 the DFAs are
 equivalent,
 i.e. the accept
 states are identical

$$A = B \text{ iff}$$

$$A \subseteq B \text{ and } A \supseteq B$$

Regular Operations (on languages)

\hookrightarrow

maintain regularity.
 Σ -alphabet, $A, B \subseteq \Sigma^*$

1. Union:

$$A \cup B = \{w \in \Sigma^* : w \in A \text{ or } w \in B\}$$

2. Concatenation:

$$AB = \{w \in \Sigma^* : w = uv, \text{ for some choice of } u \in A \text{ and } v \in B\}$$

3. Kleene Star:

IF $A \subseteq \Sigma^*$ then
 $A^* = \{\epsilon\} \cup A^1 \cup A^2 \cup \dots$
 $= \bigcup_{n \in \mathbb{N}} A^n$

The idea is that A^* is the set of all concatenations of A .

Thm: The regular Languages are closed under the regular operations.

Σ alphabet, $A, B \subseteq \Sigma^*$ regular.

$\Rightarrow A \cup B$ regular,
 $A \cap B$ regular
 A^* regular.

Pf: A, B regular

\Rightarrow exists DFA

$$M_A = (Q, \Sigma, \delta_A, q_{A_0}, F_A)$$

$$M_B = (Q_B, \Sigma, \delta_B, q_{B_0}, F_B)$$

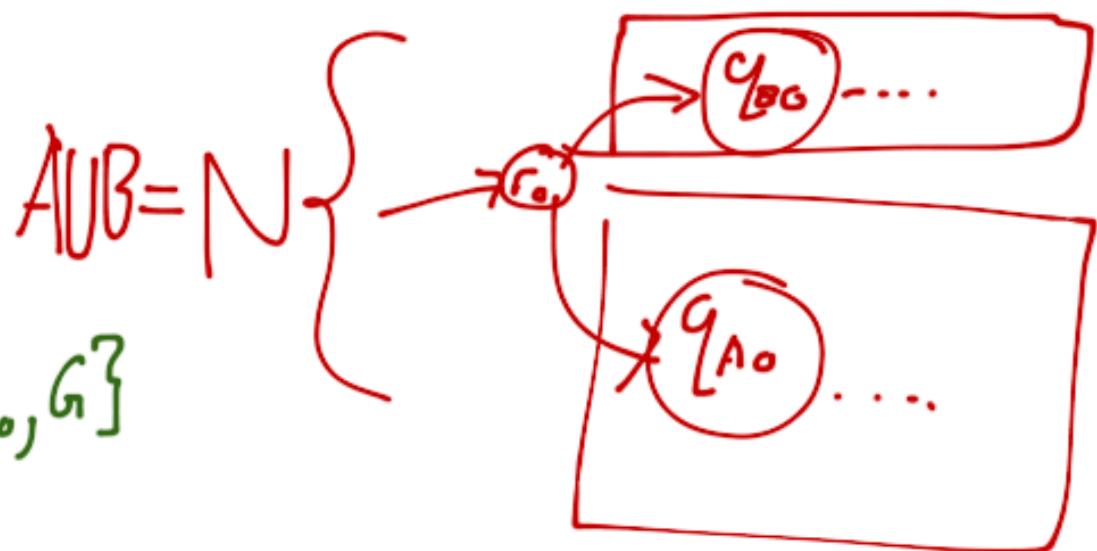
such that;

$$A = L(M_A), \quad B = L(M_B)$$

No loss of generality in assuming

$$Q_A \cap Q_B = \emptyset$$

A ∪ B:



$$N = \{R, \Sigma, \gamma, r_0, G\}$$

$$R = \{r_0\} \cup Q_A \cup Q_B$$

$$G = F_A \cup F_B$$

$$\gamma(q_A, \sigma) = \bigcup_{q_A \in Q_A} \Sigma \delta(q_A, \sigma) \quad \left[\begin{array}{l} \text{this is for } m_A, \\ m_B \text{ is similar.} \end{array} \right]$$

$$\gamma(r_0, \sigma) = \{q_{A0}, q_{B0}\}$$

$$\gamma(r_0, \sigma) = \emptyset$$

The idea is that we create γ to be the transition F_n in N to map to both A and B

Alternatively, the cartesian Product
construction:

$$M = (Q_A \times Q_B, \Sigma, \mu, (q_{A0}, q_{B0}), G)$$

$$G = \{(q_A, q_B) : q_A \in F_A \text{ or } q_B \in F_B\}$$

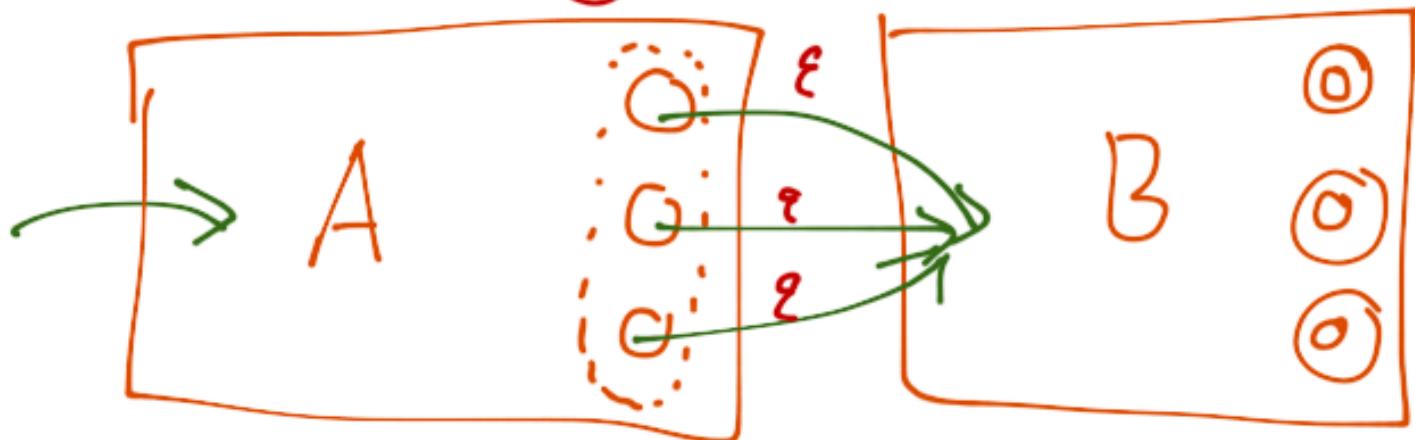
$$M = ([q_A, q_B], \sigma) = (\delta(q_A, \sigma), \delta(q_B, \sigma))$$

~~✓~~

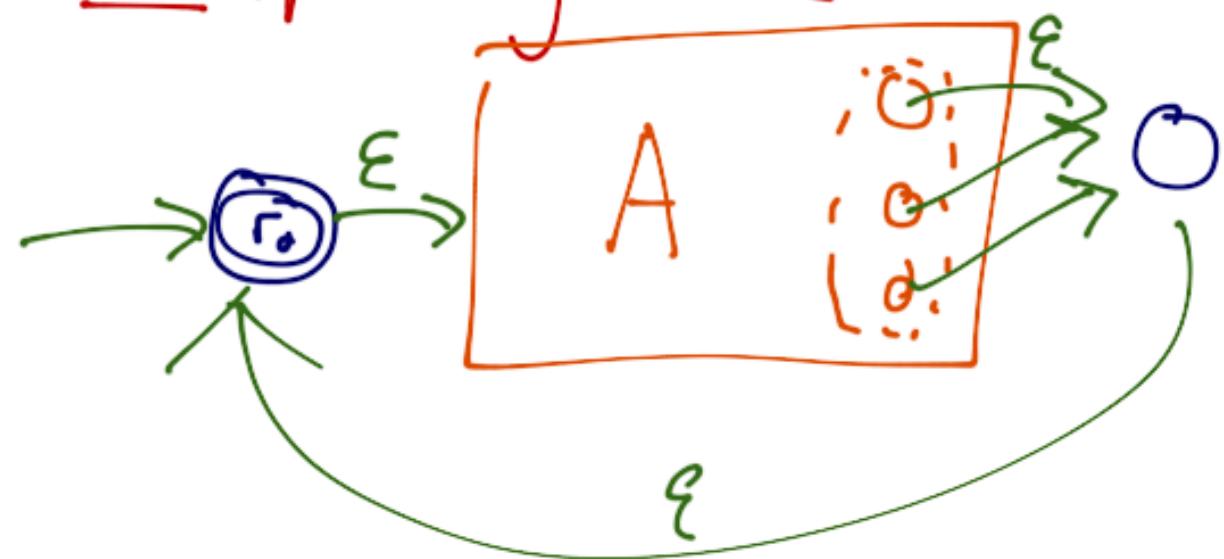
.

↖

AB: (proof by picture)



A*: (proof by Picture)



Assignment 1:

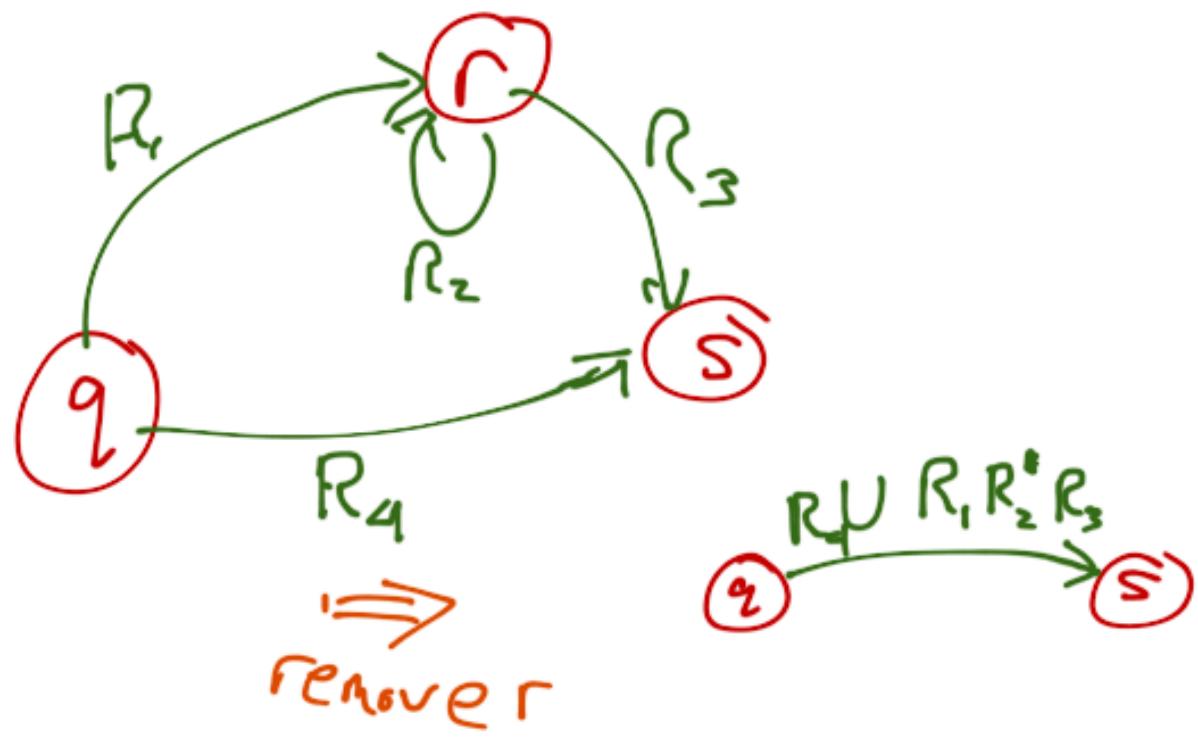
1. a) If we define $g: \mathbb{N} \rightarrow B$ as the same function as $f: \mathbb{N} \rightarrow A$ in the assumption, satisfying the requirement of the onto function. From this we can draw the conclusion that the subset of any countable set is also countable.

b)

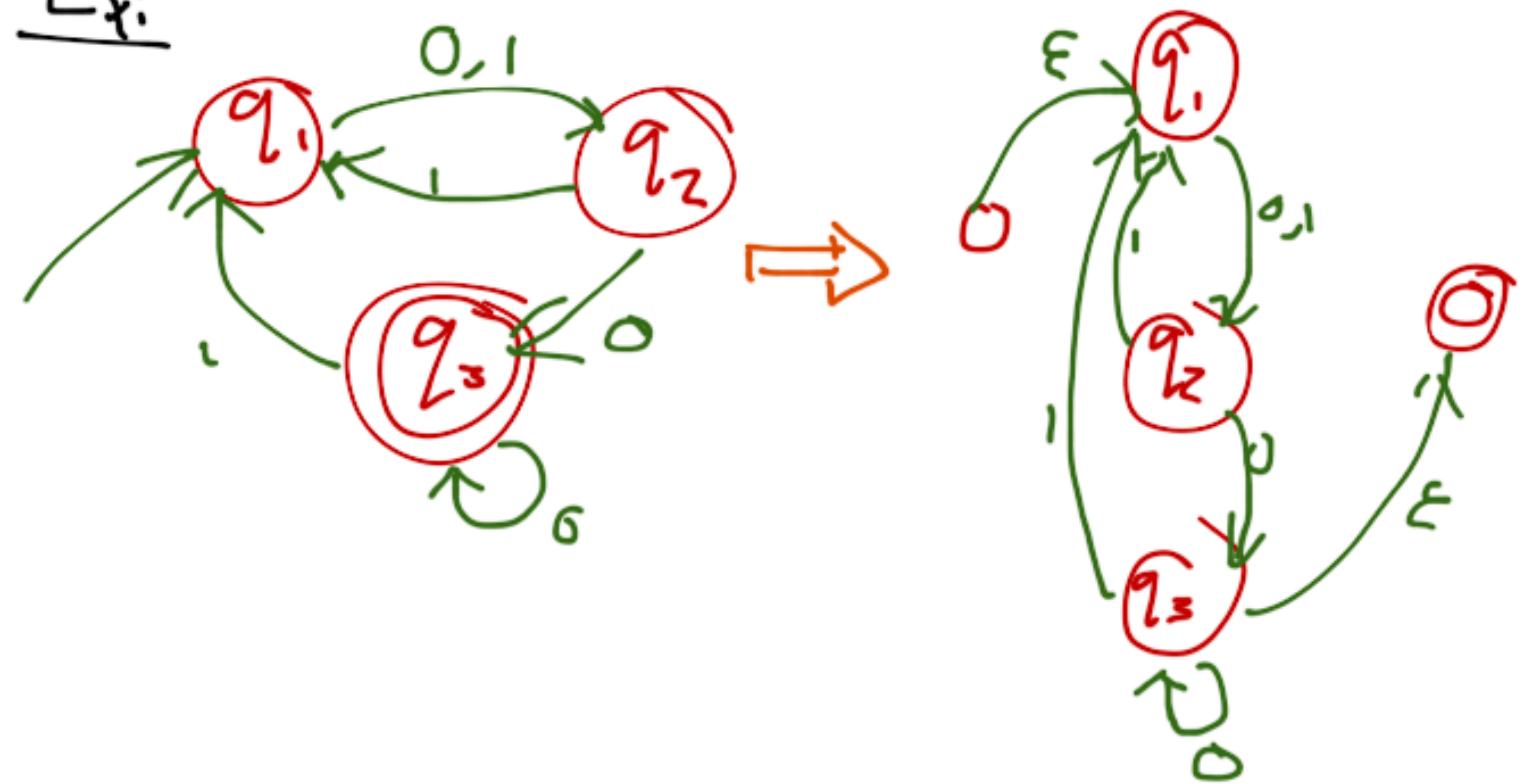
Left

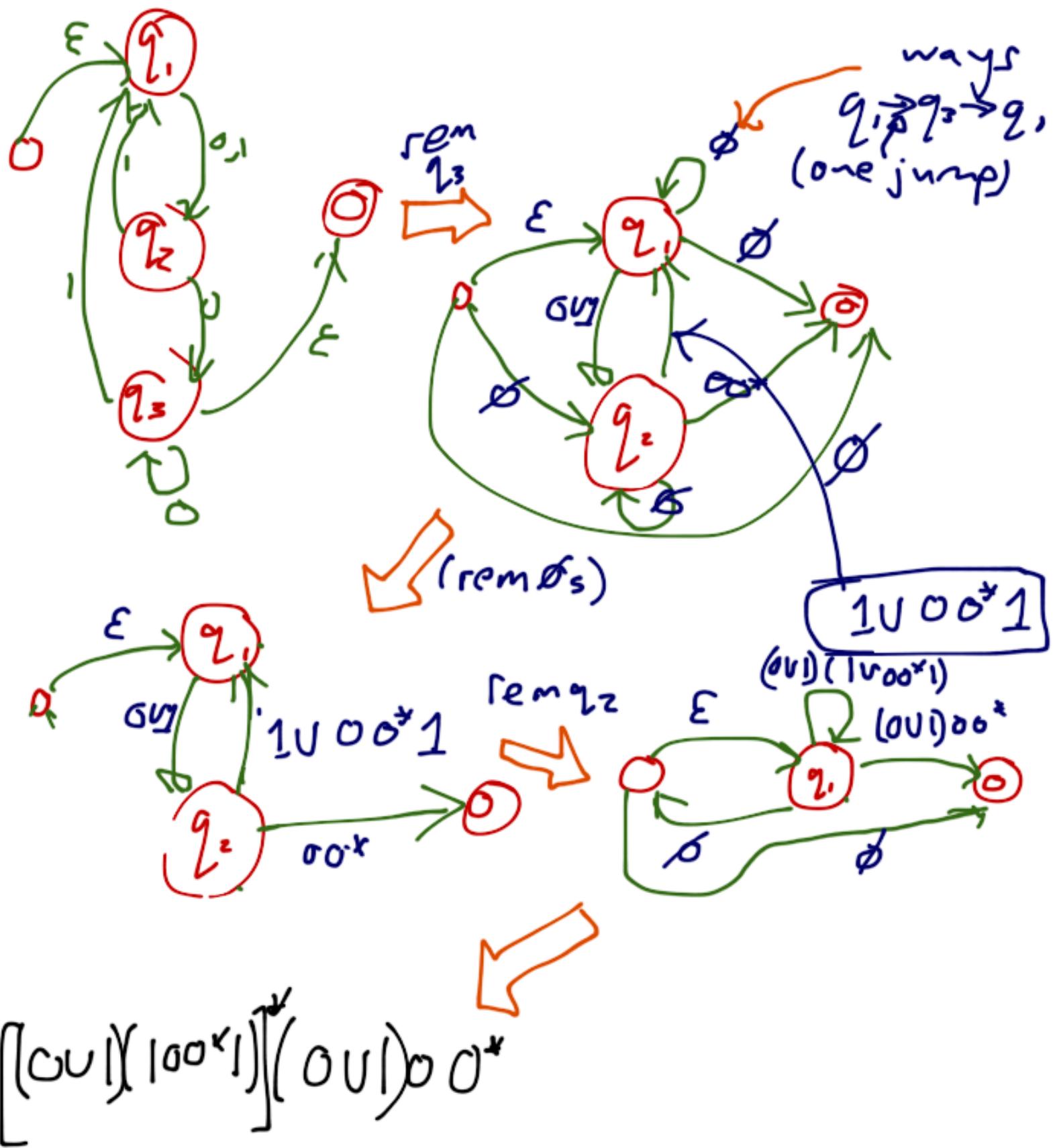
Blank

(usec)



E_x :





Techniques for proving certain languages are not regular:

Pumping Lemma:

If you have a DFA with P (or fewer) states, and you read a string with at least P symbols, you must have visited the same state twice.

If you have a string where $\text{len}(\text{str}) \geq \text{len}(\text{states})$ there exists a shorter string that arrives at the same state.

Supp. $A \in \Sigma^*$ is regular. There exists an integer $p \geq 1$ (pumping length of FA) that satisfies this property:

For every $w \in A$ with $|w| \geq p$ there exists strings $x, y, z \in \Sigma^*$ such that

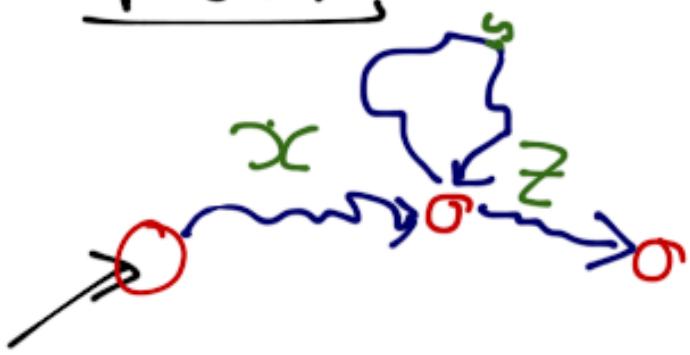
1. $y \neq \epsilon$

2. $|xy| \leq p$

3. $xy^k \in A$, for all $k \in \mathbb{N}$.

states

Proof: (sketch)



Pumping Lemma (YT video)

If L is a regular language.
 $\exists n \Rightarrow \# states \text{ in machine for } L$

$\forall z \in L$, where $|z| \geq n$

$\exists v, w, x$ such that $z = vwx$, such that $|vw| \leq n$
and $|w| \geq 1$

Then: $\forall i \geq 0$, vwx^i is also in L



The idea is that we want to disprove a language from being regular, and can do that by coming up with a z that satisfies some values of i , then using those to disprove the language of being regular.

regular \rightarrow pumping

\neg pumping \rightarrow \neg regular

¬ Pumping Property:

$\forall n$ where $n = \# \text{ states in language } L$

$\exists z \in L, |z| \geq n$

$\forall v, w, x \text{ where } z = vwx$

and $|vw| \leq n$

and $|w| \geq 1$

$\exists i \geq 0$ such that vwx^i is not in L

$\Rightarrow L$ is not regular

Ex: $A = \{0^n 1^n : n \in \mathbb{N}\}$
 $= \{0, 01, 001, 00011, \dots\}$

Claim: A is not regular.

Proof. Assume A is regular, toward contradiction.
 By the Pumping Lemma, there exist
 a pumping length p for A .

Let $w = 0^p 1^p$

We have $w \in A$ and $|w| = 2p \geq p$.
 There must therefore exist x, y, z such that:

1. $y \neq \epsilon$

2. $|xy| \leq p$

3. $xy^t z \in A$ for all $t \in \mathbb{N}$

$xyz = 0^p 1^p$
 $|xyz| \leq p$
 $y \neq \epsilon$

$\left. \begin{array}{l} |xyz| \leq p \\ y \neq \epsilon \end{array} \right\} \Rightarrow y = 0^k$ for some choice of
 $k \in \{1, \dots, p\}$



$\exists y z \in A$ for all $i \in \mathbb{N}$.

In particular, for $i = 0$:

$$\exists y^0 z = \cancel{xy} z \in A \quad \checkmark$$

but $\exists z = 0^{p-k} 1 \in A \times$

Therefore A is not regular.

□



Example:

$B = \{\omega \in \{0,1\}^*: B \text{ doesn't have the same # of } 0's \neq 1's\}$

$01|01 \in B \epsilon, 01001 \notin B$

Assume B is regular (if by contra.)

$\Rightarrow \bar{B}$ is regular

$\Rightarrow \bar{B} \cap L(0^* 1^*)$ is regular.

But, \nwarrow language of

$\bar{B} \cap L(0^* 1^*) = A$, which is not regular.
 \uparrow Pumping lemma

Example: $C = \{ w \in \{0,1\}^*: \underline{\underline{w=w^R}} \}$

$\hookrightarrow w$ is a palindrome

Claim: C is not regular.

Proof: Assume toward contradiction that C is regular.

By the PL, there is a pumping length p for C .

Let $w = 1^p 0 1^p$

we have $w \in C$ and $|w| = 2p+1 \geq p$.

Therefore, by the PL, it is possible to write $w = \cancel{x}y z$
such that.

1. $y \neq \epsilon$

2. $|x y| \leq p$

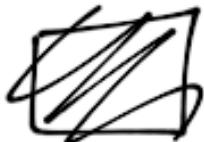
3. $x y^c z \in C$ for all $c \in \mathbb{N}$.

$$\left. \begin{array}{l} y \neq \varepsilon \\ |y| \leq p \\ xy = 1^p \end{array} \right\} \Rightarrow y = 1^k \text{ for } 1 \leq k \leq p$$

$xyz \in C$ for all $i \in \mathbb{N}$; in particular

$$xy^{4^2}z \in C$$

$$\text{But } xy^{4^2}z = 1^{p+411^k} \in 1^p \notin C$$



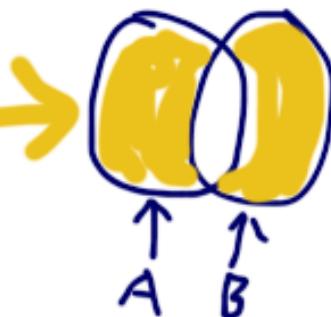
Further Discussion of Regular Languages:

A, B are sets.

→ The symmetric difference of A and B is

$$A \Delta B = (A \setminus B) \cup (B \setminus A)$$

$$(A \cup B) \setminus (A \cap B)$$



✓

Q1: $A \subseteq \Sigma^*$ is regular,

$B \subseteq \Sigma^*$ is ; it necessarily the case that $A \Delta B$ is regular?

Yes, B is finite $\Rightarrow B$ is regular

B is regular $\Rightarrow \bar{B}$ is regular,

A regular $\Rightarrow \bar{A}$ regular

A regular, \bar{B} regular $\Rightarrow A \cap \bar{B}$ regular

\bar{A} regular, B regular $\Rightarrow \bar{A} \cap B$ regular

$A \cap \bar{B}$ regular, $\bar{A} \cap B$ regular $\Rightarrow (A \cap \bar{B}) \cup (\bar{A} \cap B)$ regular

$A : W \{w, \dots, w\} \subseteq A, X \{x, \dots, x\} \subseteq A.$

$$(A \setminus W) \cup X = A \Delta (W \cup X)$$

Q1b. $A \subseteq \Sigma^*$ is not regular. $B \subseteq \Sigma^*$ finite.
 $\Rightarrow A \Delta B$ non-regular

Assume for Contradiction $A \Delta B$ regular.

$\Rightarrow (A \Delta B) \Delta B$ is regular

But $(A \Delta B) \Delta B = A$, which is not regular.

Q1c: $A \subseteq \Sigma^*$, $B \subseteq \Sigma^*$, (neither are regular)
Is it necessarily that $A \Delta B$ is non-regular?

False.

If $A = B = \{0^n 1^n \mid n \in \mathbb{N}\}$, $A \Delta B = \emptyset$, which is regular.

Qz: Supp. $A \subseteq \Sigma^*$ is regular.

Def: $A^R = \{w^R : w \in A\}$. Is A^R necessarily regular?

Have a DFA "M" for A.

$$M = (Q, \Sigma, \delta, q_0, F)$$

Define NFA "N"

$$N = (Q \cup \{r_0\}, \Sigma, \gamma, r_0, \{q_0\})$$

$Q \cup \{r_0\}$

$$\begin{aligned} \gamma(r_0, \epsilon) &= F \\ \gamma(q, \sigma) &= \{p \in Q : \delta(p, \sigma) = q\} \\ \forall q \in Q, \sigma \in \Sigma \end{aligned}$$

$A \text{ regular} \Leftrightarrow \exists \text{ reg. expr'n } R \text{ for } A.$

Supp S. is reg expr:

if $s = \emptyset, s = \epsilon, s = \sigma, s^R = s$

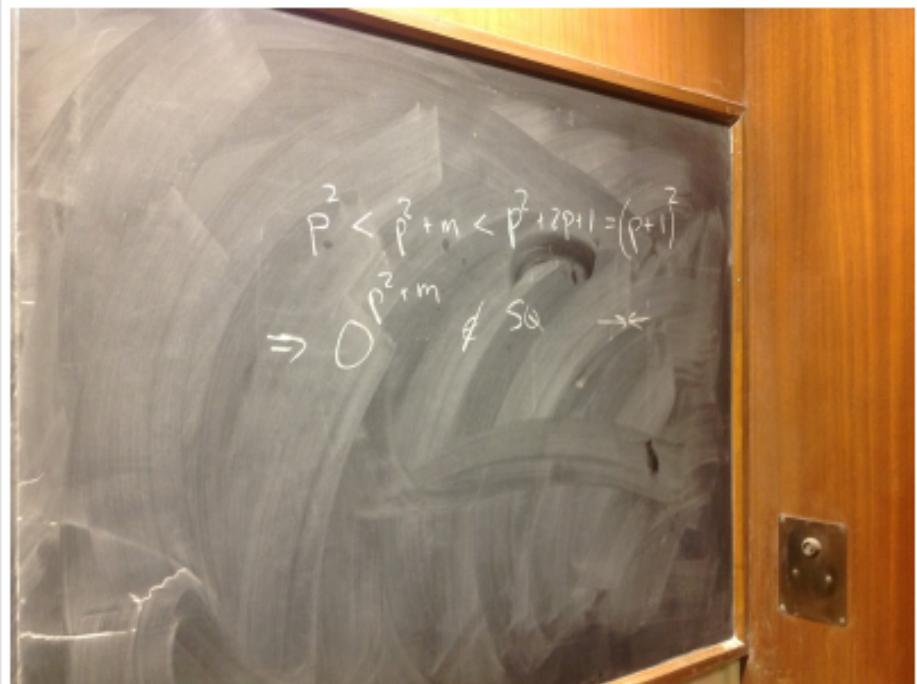
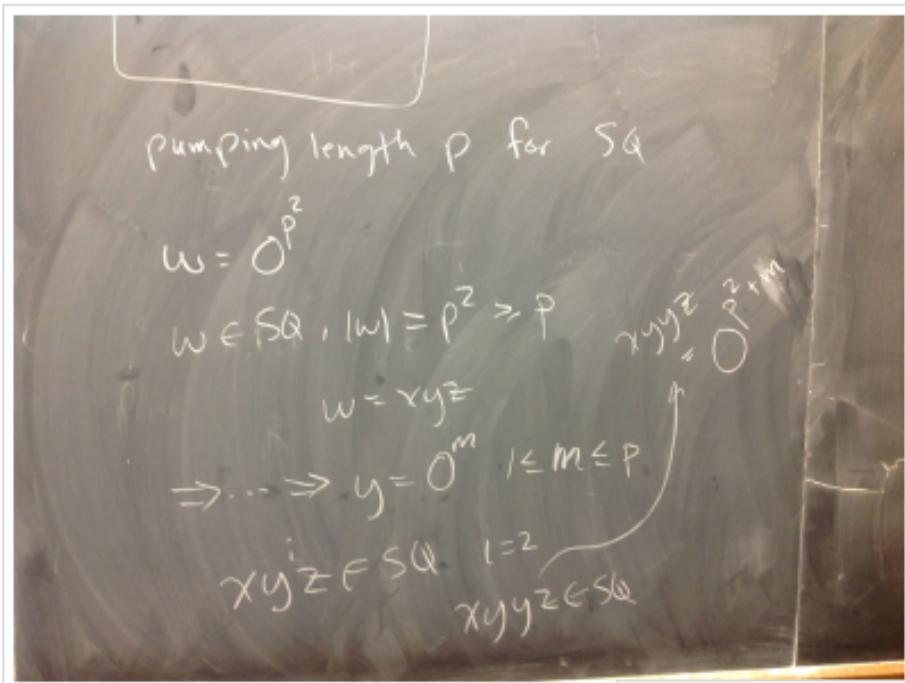
if $\begin{cases} s = s_1 \cup s_2 & \Rightarrow s^R = (s_1^R \cup s_2^R) \\ s = s_1 s_2 & \Rightarrow s^R = (s_1^R s_2^R) \\ s = s_1^* & \Rightarrow s^R = (\Sigma_1^R)^* \end{cases}$

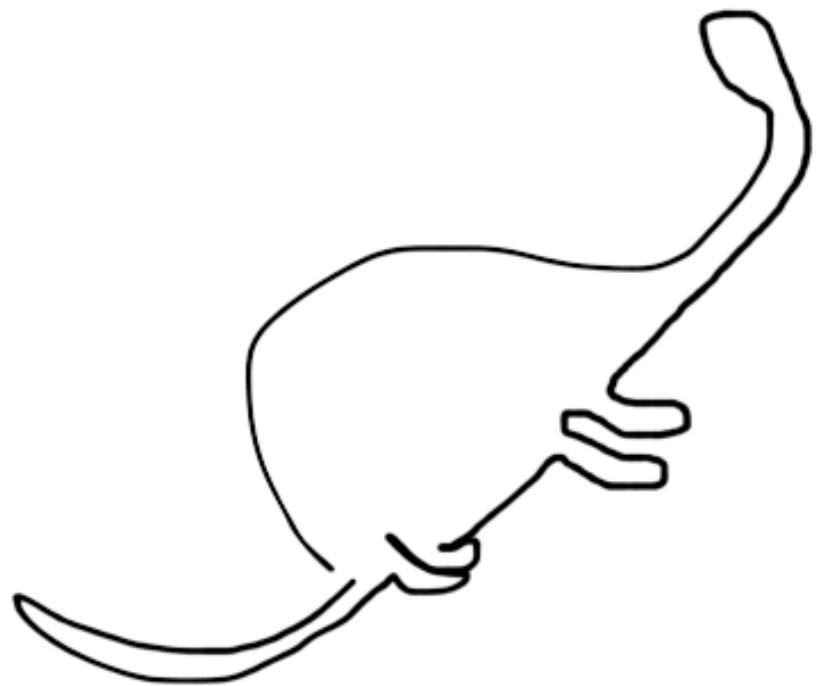
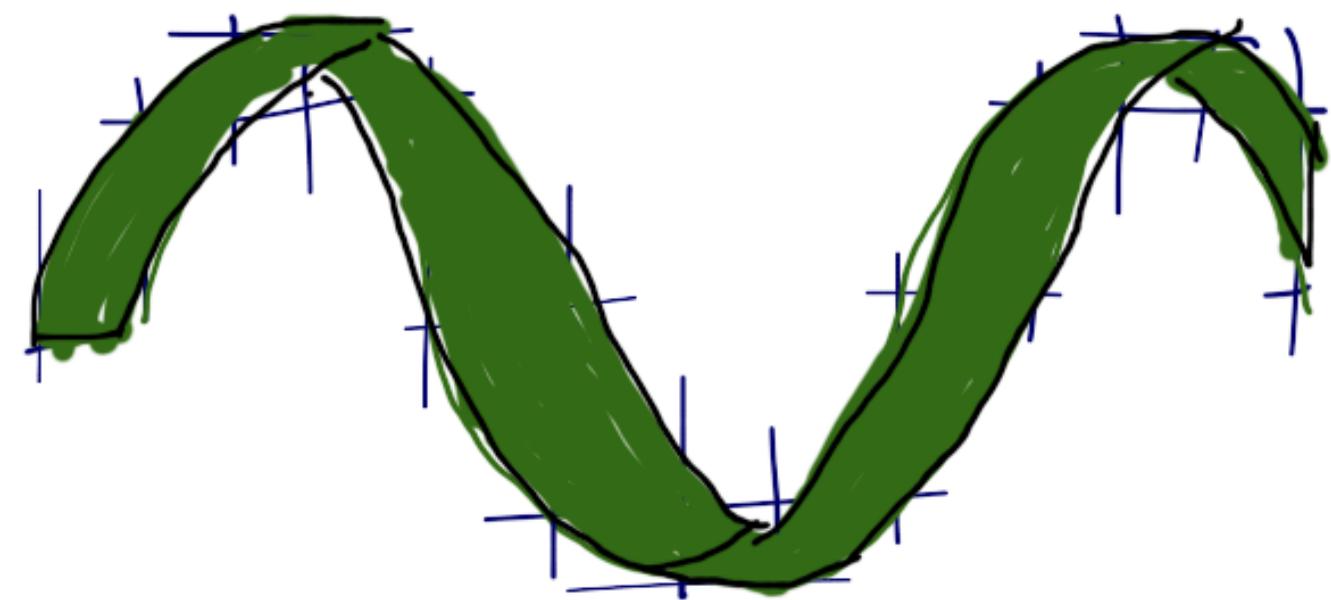
Exercise: do Q2 with reg. expr's

$$\Sigma = \{0\}$$

$SQ = \{0^n : n \text{ is a perfect square}\}$

Claim: SQ is not regular.





Context-Free Grammars:

A 4-Tuple of:

$$G = (V, \Sigma, R, S)$$

where:

1. V is a finite, nonempty set of variables
2. Σ is an alphabet ($V \cap \Sigma = \emptyset$)
3. R is a finite set of rules of the form:

$$A \xrightarrow{} w$$
$$A \in V, w \in (V \cup \Sigma)^*$$

4. $S \in V$ is the start variable

ex:

$$G = (V, \Sigma, R, S)$$

$$V = \{S\}$$

$$\Sigma = \{0, 1\}$$

R includes these 2 rules:

$$S \xrightarrow{} 0S1$$

$$S \xrightarrow{} \epsilon$$

Every Grammar Generates a language:
 $L(G) \subseteq \Sigma^*$

Informally:

1. write down start variable $S \rightarrow$ current str
2. loop:
 Pick any var in the current string
 with one of its expansions according to
 rules R of the C.F.L.
3. if you are left with $x \in \Sigma^*$,
 then x is generated by G .

Definition: Let $G = (V, \Sigma, R, S)$ be a CFG

1. For all $u, v, w \in (V \cup \Sigma)^*$ and $A \in V$, we have:
 $uAr \Rightarrow^* uvwv$, if $A \rightarrow w \in R$
2. For $x, y \in (V \cup \Sigma)^*$, we have $x \Rightarrow^* y$ *
if there exists $K \geq 1$ and $w_1, \dots, w_K \in (V \cup \Sigma)^*$
such that $x \Rightarrow_G w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_K \Rightarrow_G y$
3. $L(G) = \{x \in \Sigma^*: S \Rightarrow^* x\}$
 $S \Rightarrow_G w_1 \Rightarrow w_2 \dots \Rightarrow_G w_K = x$
is called a derivation of x
4. A language $A \subseteq \Sigma^*$ is context-free iff
there exists a CFG G such that $A = L(G)$

Ex: $\{x \in \{0, 1\}^*: x = x^R\}$

$$\begin{array}{l} S \Rightarrow 0S0 \\ S \Rightarrow 1S1 \\ S \Rightarrow 0 \\ S \Rightarrow 1 \\ S \Rightarrow \epsilon \end{array}$$

Shorthand

$$S \Rightarrow 0S0 \mid 1S1 \mid 1 \mid 0 \mid \epsilon$$

Definition:

$$\text{Supp: } M = (Q, \Sigma, \delta, q_0, F)$$

is a DFA and $w \in \Sigma^*$. The DFA M accepts w iff

1. $w = \epsilon$, and $q_0 \in F$, or
2. $w = q_0 \dots q_n$ for $n \geq 1$, $q_1 \dots q_n \in \Sigma$ and
there exist states $r_0 \dots r_n \in Q$
such that $r_0 = q_0$, $r_n \in F$ and
 $\delta(r_k, q_{k+1}) = r_{k+1}$ for all $k \in \{0 \dots n-1\}$

An equivalent def:

- extend δ to operation $r \in \Sigma^*$.

Define:

$$\delta^*: Q \times \Sigma^* \rightarrow Q$$

as follows:

1. $\delta^*(q, \epsilon) = q$
2. $\delta^*(q, w\sigma) = \delta(\delta^*(q, w), \sigma)$ for all $q \in Q$, $w \in \Sigma^*$,
 $\sigma \in \Sigma$

M accepts w iff $\delta^*(q_0, w) \in F$

Ex:
G:

$$S \rightarrow 0S1S \mid 1S0S \mid \epsilon$$

$$L(G) = ?$$

$$A \stackrel{\text{def}}{=} \{x \in \{0, 1\}^*: |x|_0 = |x|_1\}$$

$$\boxed{L(G) = A} \text{ claim}$$

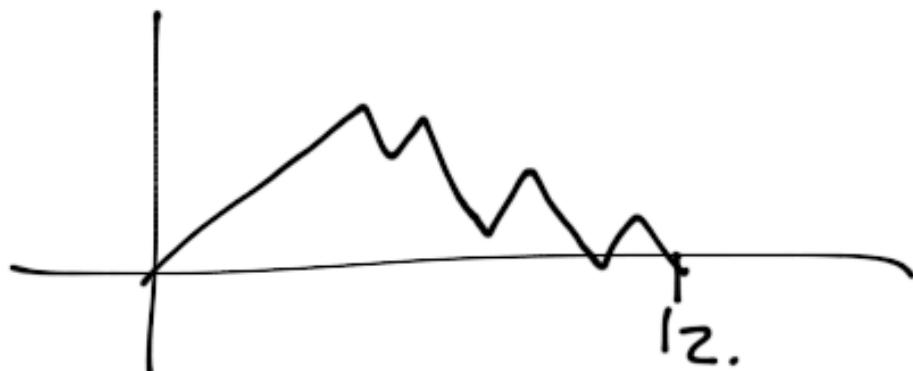
$$L(G) \subseteq A$$

clear from inspection of the grammar

$$A \subseteq L(G)$$

$$\text{Supp } x \in A$$

$$000101b1101$$



G: $s \rightarrow (s) | ss | \epsilon$
 $\Sigma = \{\langle , \rangle\}$

$s \xrightarrow{\zeta} ss \xrightarrow{\zeta} (s)s \xrightarrow{\zeta} ((s))s \xrightarrow{\zeta} ((())s$

$\Rightarrow (())(s) \xrightarrow{\zeta} (())()$ $\Rightarrow (())() \in L(G)$

Likewise,

$\{(), (()), ()(), ((())), \dots\} \in L(G)$
 $\}, >(), ((), \dots) \notin L(G)$

$L(G)$ is the language of balanced parens.

This is an ex. of Leftmost Derivation
we can always derive strings in $L(G)$
(\forall grammars G) using a leftmost derivation.

$$\Sigma = \{a, b, +, *, (,)\}$$

$$S \rightarrow S+S \mid S*S \mid (S) \mid a \mid b$$

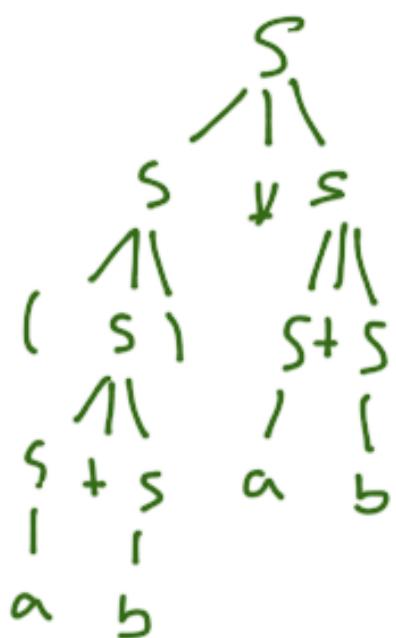
} "R"

$(a+b)*a+b$

$$\begin{aligned} S &\Rightarrow S * S \Rightarrow (S+S) * S \Rightarrow (a+S) * S \Rightarrow (a+b)*S \\ &\Rightarrow (a+b)*S+S \Rightarrow (a+b)*a+S \Rightarrow (a+b)*a+b \end{aligned}$$

This grammar generates all formulas over $a, b, +, *$ and parens

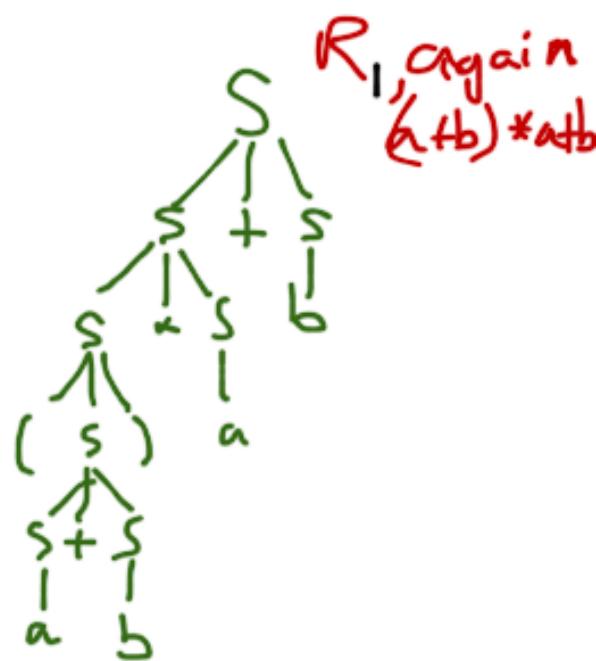
Parse Tree:



There is a one-to-one and onto (bijection?) between parse trees & leftmost derivations.

A CFG G is ambiguous if
 \exists string s , $s \in L(G)$ having
 2 or more parse trees.

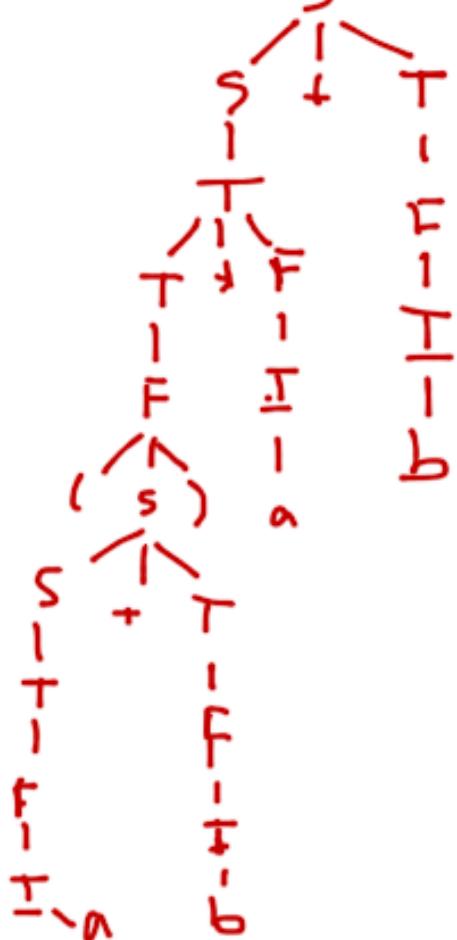
For many CFG applications
 we need unambig. grammars,



~~Try again~~ Try again w/ R_2 : (being unambiguous now)

$$\begin{aligned} S &\rightarrow T \mid S+T \\ T &\rightarrow F \mid T*F \\ F &\rightarrow I \mid (S) \\ I &\rightarrow a \mid b \end{aligned}$$

unique parse tree $(a+b)^*ab$



Proung that various
 grammars are/aren't
 (un)ambiguous very
 hard.

CFL v/no Unimbig CFG:

$\{0^n 1^m 0^k : n=m \text{ or } m=k\}$

$$\begin{array}{l} S \rightarrow A|B \\ A \rightarrow C D \\ C \rightarrow 0 C 1 | \epsilon \\ D \rightarrow 0 D | \epsilon \\ B \rightarrow D E \\ E \rightarrow 1 E 0 | \epsilon \end{array}$$

$\{0^n 1^m : n \neq m\}$

$$\begin{array}{l} S \rightarrow 0 S | A | B \\ A \rightarrow 0 A | 0 \\ B \rightarrow B | 1 | 1 \end{array}$$

$A = \{0^n 1^n : n \in \mathbb{N}\}$

Show A^* is context-free.

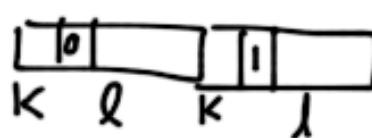
$$\begin{array}{l} S \rightarrow A S | \epsilon \\ A \rightarrow 0 A 1 | \epsilon \end{array}$$


A CFL A having no unambiguous
CFG is said to be inherently ambiguous

$L_2 = \{uv: u, v \in \{0, 1\}^*, |u|=|v|, u \neq v\}$

$S \rightarrow 0A1 \quad | A0 \quad | \text{OSO} \quad | \text{ISI}$
 $A \rightarrow XAX \quad | \text{2}$
 $X \rightarrow 0 \quad | \quad 1$

$L_3 = \{uv: u, v \in \{0, 1\}^*\}$


 $\rightarrow x^k 0 x^{k+l} 1 x^l$

$S \rightarrow AB \quad | BA$
 $A \rightarrow XAX \quad | 0$
 $B \rightarrow XBx \quad | 1$
 $X \rightarrow 0 \quad | 1$

Closure Props of CFLs:

Thru Σ alphabet, $A, B \subseteq \Sigma^*$ are context-free languages.

\Rightarrow These languages_↓ are also context-free:

$$A \cup B, A \cap B, A^*$$

(The CFL's are closed under regular oper.s)

Pf: Let $G_A = (V_A, \Sigma, R_A, S_A)$
 $G_B = (V_B, \Sigma, R_B, S_B)$

be CFGs, w/ $A = L(G_A) \subsetneq B = L(G_B)$

w/o loss of generality (wLOG), assume
 $V_A \cap V_B = \emptyset$ and $S \notin V_A \cup V_B$

CFG for:

$A \cup B$:

$S \rightarrow S_A | S_B$

$A B$:

$S \rightarrow S_A S_B$

A^*
 $S \rightarrow S_A S | \epsilon$

$L = (V, \Sigma, R, S)$

$V = V_A \cup V_B \cup \{S\}$

$R = R_A \cup R_B \cup \{S \rightarrow S_A | S_B\}$

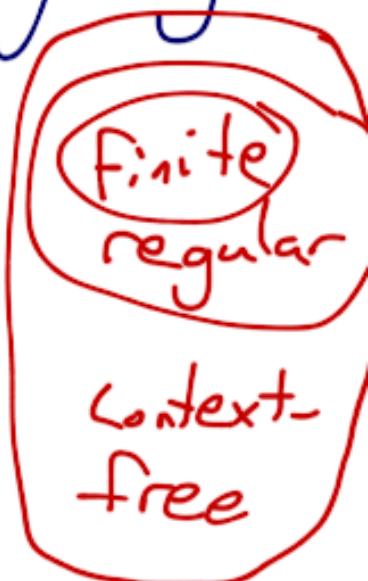
Thm: Suppose Σ is an alphabet, $A \subseteq \Sigma^*$ is a regular language.

$\Rightarrow A$ is a context-free language.

Pf. 1 (using regular expr)

Let R be a regular expr.
for A .

We will define a CFG G
s.t. $L(G) = L(R)$



Left for FB interview

Proving some languages are not context free:

Pumping Lemma For CFLs:

$\forall \Sigma$ and $\forall A \subseteq \Sigma^*$

\exists pumping length $p \geq 1$ for A
where $\forall w \in A$: $|w| \geq p$ and

it is impossible to write $w = uvwxyz$ for
 $u, v, w, x, y, z \in \Sigma^*$ such that:

1. $v, y \neq \epsilon$

2. $|vxy| \leq p$

3. $uv^i xy^i z \in A, \forall i \in \mathbb{N}$

Pf:

Given that A is Context-Free, there exists a CFG G such that $A = L(G)$

Assume WLOG that G is in Chomsky normal form.

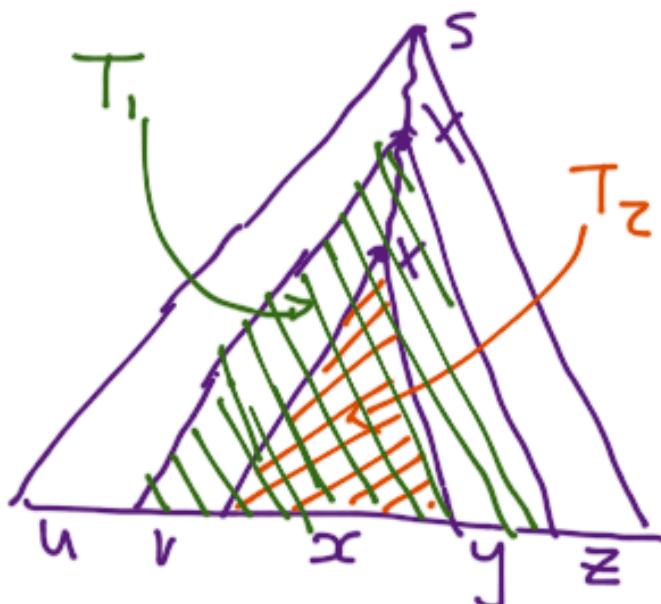
let m be the number of variables in G , and let $p = 2^m$

Suppose $w \in A$ $|w| \geq p = 2^m$. G is in Chomsky normal form, so any parse tree for w has $2|w|-1$ internal (variable) nodes

$$2|w|-1 \geq 2 \cdot 2^m - 1 > 2^m$$

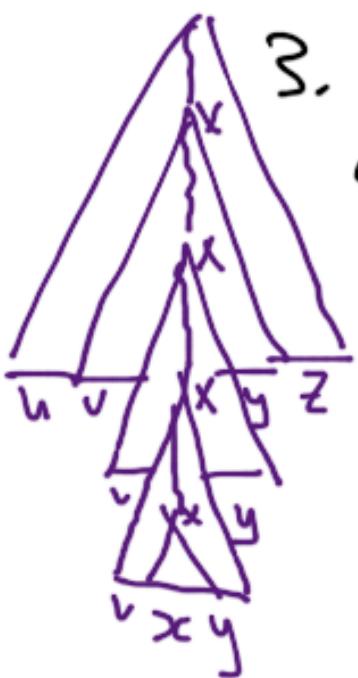


There must be a path from the root to a leaf with at least $m+1$ variable nodes.



we define u, v, x, y, z as
the picture suggests

1. $vy \notin \Sigma$, otherwise
we could build a
strictly smaller parse
tree for w ...
impossible in CNF.



3. Continue this way to obtain
a parse tree for
 $uv^ixy^iz, i \in \mathbb{N}$.



2. we want to choose T_1 as small
as possible to prove
 $|vxy| \leq p$

Using the pumping Lemma For CFG's (ex:)

$$A = \{0^n 1^n 2^n : n \in \mathbb{N}\}$$

$$\Sigma = \{0, 1, 2\}$$

Claim: A is not Context-free

Proof: Assume towards contradiction that A is context-free.

By the PL for CFLs, there exists a P for A .

Let $w = 0^p 1^p 2^p$. We have $w \in A$ and $|w| = 3p \geq p$. So there must exist $u, v, y, z \in \Sigma^*$ such that

$$\rightarrow w = u v x y z$$

$$\rightarrow v y \neq \epsilon$$

$$\rightarrow |v y| \leq p$$

$$\rightarrow u v^i x y^i z \in A \text{ for all } i \in \mathbb{N}.$$



$$\omega = \sigma_1 \rho_2^p = uvxyz$$

$|vxy| \leq p$, so it is impossible for all 3 symbols σ_1, ρ_2 to appear within vxy

$Vy \neq \epsilon$, so at least one of the 3 symbols must appear within vy .

So, taking $i=0$, we know that $uv^0x^0y^0z$ must contain strictly fewer of one of the symbols than another.

This conflicts the statement that $uv^0x^0y^0 \in A$

→ Therefore A is not context-free.



Ex²:

$B = \{rzs; r, s \in \{0, 1\}^*, r \text{ is a substring of } s\}$

Claim B is not Context-Free

Proof: Assume towards contra. that B is context-Free.

By the PL for CFLs, there must be a pumping length $p \geq 1$ for B .

Let $\omega = 0^p 1^p 2 0^p 1^p$, $\omega \in B$, and $|w| > p$

So it is possible to write $\omega = uvxyz$ so that $1, 2, 3$ from the CFG PL holds

There are these possibilities for the location of the 2 in $uvxyz$.

Case 1: z lies within u

Consider $i=0$

$$uv^0x^0y^0z = uxz \in B.$$

We have $v \neq \epsilon$, so

$$uxz = G^p i^p 0^{p-k} i^{p-l}, \quad k+l \geq 1$$

So this is not a string in B , so we have a contradiction for this case.

Case 2: The z lies within v or y .

Consider $i=2$

$uv^2x^2y^2z$ has 2 z 's, so it is invalid.



-

Pumping Lemma for CFLs:

$(\forall A \subseteq \Sigma^*; A \text{ is a CFL}) (\exists p \geq 1) (\forall w \in A: |w| \geq p, w = uvxyz \therefore vy \neq \epsilon \wedge |vxy| \leq p \wedge (\forall i \in \mathbb{N} \ u v^i x y^i z \in A))$

$B = \{ r^2 s : r, s \in \{0, 1\}^* \text{, } r \text{ is a substring of } s \}$

Claim: B is not context-free.

Pf: Towards contradiction, assume B is context free. . . . pumping length P

let $w = 0^P 1^P 2 0^P 1^P, w = uvxyz \dots$

Case 1: $2 \in u$ [yesterday]

Case 2: $2 \in V$ or $2 \in Y$ [yesterday]



Case 3a: $z \in \Sigma$

$$uv^2xy^2z = 0^p 1^p 2 0^p 1^p$$

\uparrow_z

We have $|vxy| \leq p$

It must be that $v = 1^k$ and $y = 0^l$
 $(k+l \geq 1)$

3ai: $k > 0$ consider $i = 2$
 $uv^2xy^2z = 0^p 1^{p+k} 2 0^{p+l} 1^p \notin B$

3bi: $l > 0$ consider $i = 0$

$uv^0xy^0z = uxz = 0^p 1^{p-k} 2 0^{p-l} 1^p \notin B$

Case 4: $z \notin \Sigma$

consider $i = 2$

$uv^2xy^2z \notin B$ because it has
 the form $r 2 0^p 1^p$ where $|r| > 2p$.

~~⇒ B is not context free~~

CFLs are Not closed under Intersection.

Ex: $A = \{0^n 1^n z^k : n, k \in \mathbb{N}\}$

$$S \rightarrow X Y$$

$$X \rightarrow 0 X_1 | \epsilon$$

$$Y \rightarrow 2 Y_1 | \epsilon$$

A is context-free.

$$B = \{0^K 1^n z^n : n, K \in \mathbb{N}\}$$

B is context-free.

$$A \cap B = \{0^n 1^n z^n : n \in \mathbb{N}\}$$

which is not context-free

CFLs are not closed under complement.
Follows DeMorgan's Laws

Ex: $C = \{0^n 1^n 2^n : n \in \mathbb{N}\}$

prove \bar{C} is context-free

CFG for \bar{C} :

$S \rightarrow X Y_2 | Y_0 W | U_1 O U | U_2 O U | U_2 I U$

$X \rightarrow 0 X | Y_0 O | Y_1 I$

$Y_0 = 0 Y_0 | \epsilon$

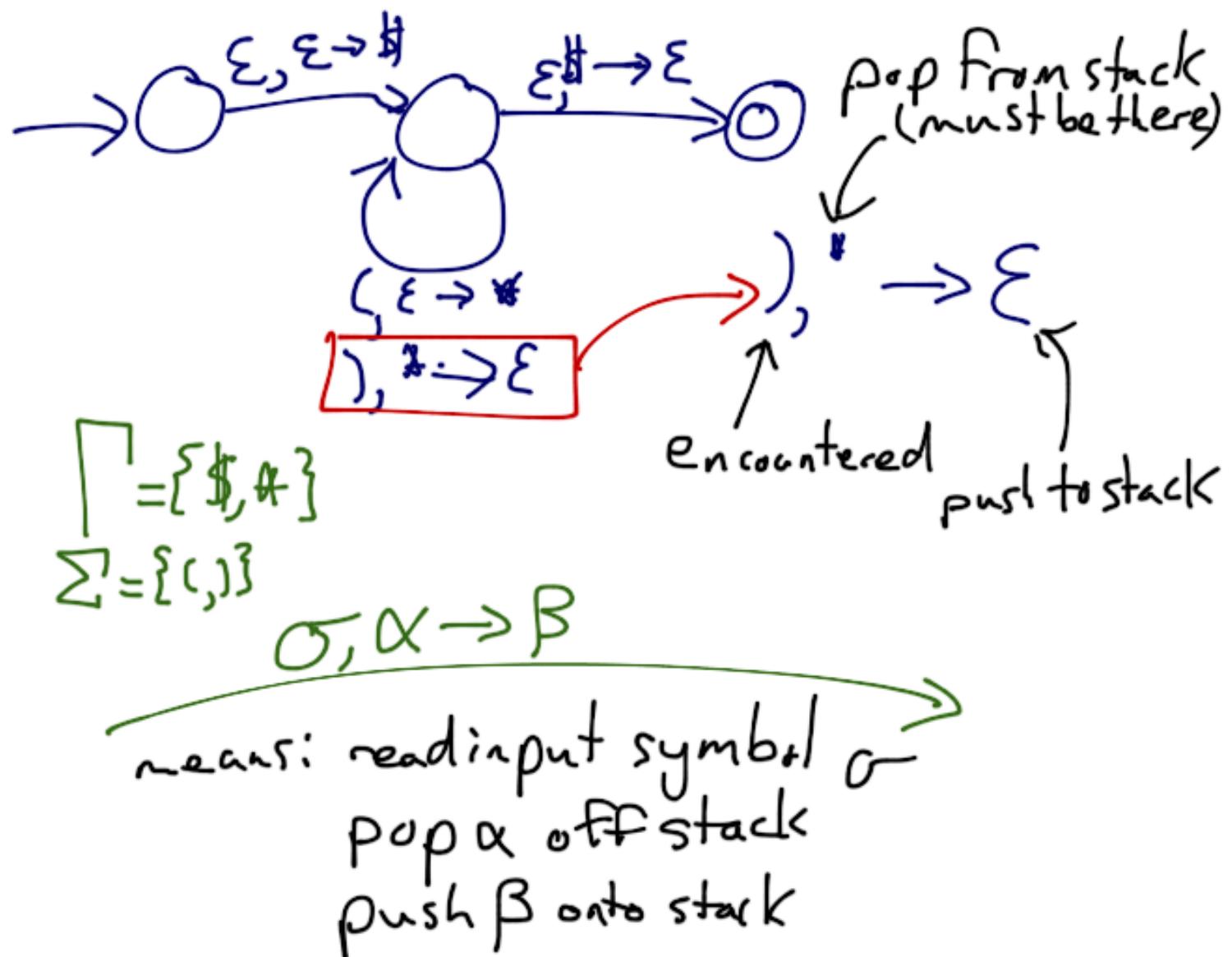
$Y_1 = Y_1 | \epsilon$

$Y_2 = 2 Y_2 | \epsilon$

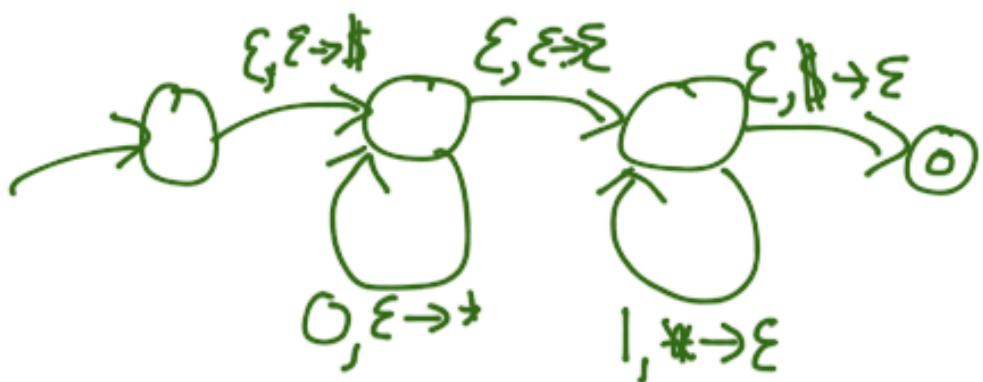
$U \rightarrow O U | I U | Z U | \epsilon$

$W \rightarrow I W Z | Y_1 | Y_2 Z$

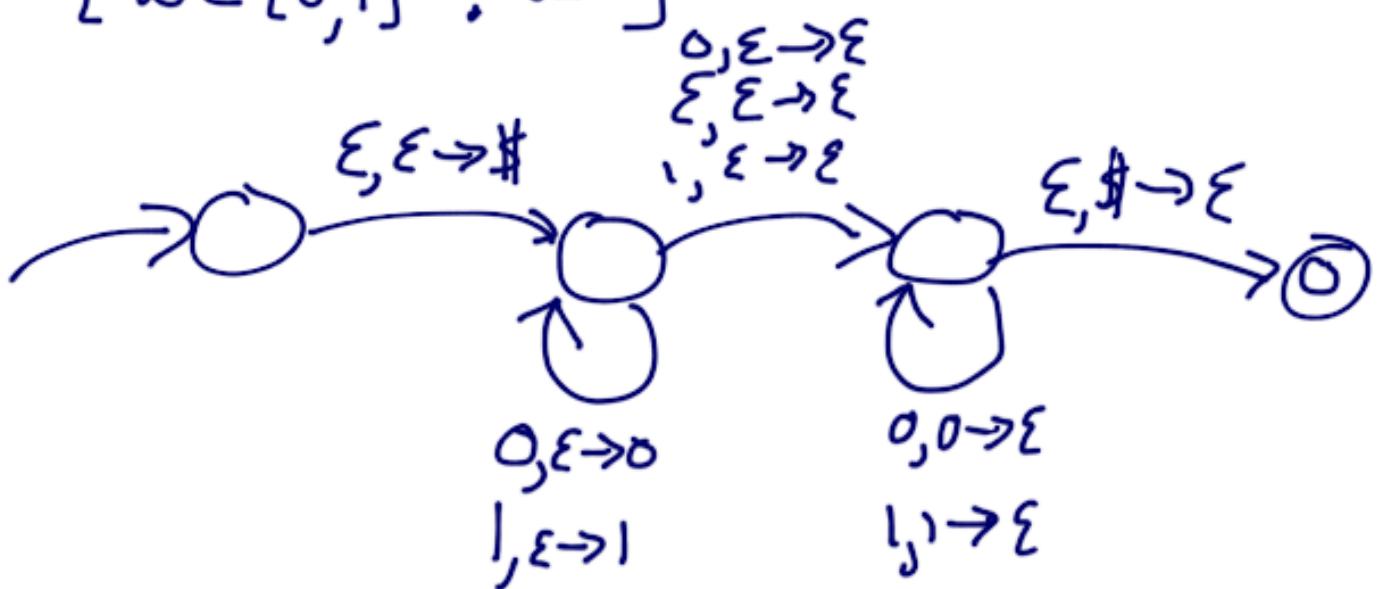
Pushdown Automaton: NFA's with 1 stack



$\{\sigma^n : n \in \mathbb{N}\}$



$\{w \in \{0, 1\}^* : w = w^R\}$



Thm $\Sigma \text{pp} A \subseteq \Sigma^*$

A is CF iff $A = L(P)$ for some P