BILL AND NATHAN START RECORDING

Review for the Midterm

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.
3. **Caveat** You must hand in your own work and you must understand what you hand in.

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.
3. **Caveat** You must hand in your own work and you must understand what you hand in.
4. **Warning** Mindlessly copying does not work.

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.
3. **Caveat** You must hand in your own work and you must understand what you hand in.
4. **Warning** Mindlessly copying does not work.
5. **Neat** LaTex is best. Good handwriting okay. Draw Aut, or use LateX tool posted.

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.
3. **Caveat** You must hand in your own work and you must understand what you hand in.
4. **Warning** Mindlessly copying does not work.
5. **Neat** LaTex is best. Good handwriting okay. Draw Aut, or use LateX tool posted.
6. **Our Intent** This is exam I intended to give out originally. The extra time is meant for you to format and put in LaTeX.

Rules

1. **Begin** Midterm ON Gradescope: Tuesday March 23, 8:00PM-10:00PM. (IF this is a problem for you contact me ASAP!!)
2. **Resources** Midterm is open-everything. The web, my notes, my HW solutions, all fine to use. Cannot ask any other human. Can talk to your cat. Honor System. I trust and respect you.
3. **Caveat** You must hand in your own work and you must understand what you hand in.
4. **Warning** Mindlessly copying does not work.
5. **Neat** LaTex is best. Good handwriting okay. Draw Aut, or use LateX tool posted.
6. **Our Intent** This is exam I intended to give out originally. The extra time is meant for you to format and put in LaTeX.
7. **Scope of the Exam**

   Short Answer HWs and lectures.

   Long Answer This Presentation.

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$

   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^* w$

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$

   For a fixed string w, $w\{a, b\}^*$, $\{a, b\}^* w$

2. $\{a, b\}^* a\{a, b\}^n$ (DFA: 2^{n+1}, NFA: $n + 2$, CFG: $\log n$. Cool!)

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**
   Numbers that are $\equiv i \pmod{j}$
   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$
   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^*w$
2. $\{a,b\}^*a\{a,b\}^n$ (DFA: 2^{n+1}, NFA: $n+2$, CFG: $\log n$. Cool!)
   $\{a^i : i \neq n\}$ (DFA requires $\sim n$, NFA $\sim 2\sqrt{n} + \mathrm{logstuff}$ Cool!)

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**
   Numbers that are $\equiv i \pmod{j}$
   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$
   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^*w$
2. $\{a,b\}^*a\{a,b\}^n$ (DFA: 2^{n+1}, NFA: $n+2$, CFG: $\log n$. Cool!)
   $\{a^i : i \neq n\}$ (DFA requires $\sim n$, NFA $\sim 2\sqrt{n} + \mathrm{logstuff}$ Cool!)
   Others

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$

   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^* w$

2. $\{a,b\}^* a \{a,b\}^n$ (DFA: 2^{n+1}, NFA: $n+2$, CFG: $\log n$. Cool!)

   $\{a^i : i \neq n\}$ (DFA requires $\sim n$, NFA $\sim 2\sqrt{n} + \mathrm{logstuff}$ Cool!)
   Others

3. DFA, NFA, REGEX. Equivalence of all of these.

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**
   Numbers that are $\equiv i \pmod{j}$
   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$
   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^*w$
2. $\{a,b\}^*a\{a,b\}^n$ (DFA: 2^{n+1}, NFA: $n+2$, CFG: $\log n$. Cool!)
   $\{a^i : i \neq n\}$ (DFA requires $\sim n$, NFA $\sim 2\sqrt{n} + \text{logstuff}$ Cool!)
   Others
3. DFA, NFA, REGEX. Equivalence of all of these.
4. Closure Properties.

What We Have Covered: Regular Languages

1. **Examples of Reg Langs**

   Numbers that are $\equiv i \pmod{j}$

   $\{w : \#_a(w) \equiv i_1 \pmod{j_1} \wedge \#_b(w) \equiv i_2 \pmod{j_2}\}$

   For a fixed string w, $w\{a,b\}^*$, $\{a,b\}^*w$

2. $\{a,b\}^*a\{a,b\}^n$ (DFA: 2^{n+1}, NFA: $n+2$, CFG: $\log n$. Cool!)

   $\{a^i : i \neq n\}$ (DFA requires $\sim n$, NFA $\sim 2\sqrt{n} + \mathrm{logstuff}$ Cool!)

   Others

3. DFA, NFA, REGEX. Equivalence of all of these.
4. Closure Properties.
5. Non-Regular: ZW Pumping Lemma, Closure properties.

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathsf{N}\}$

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathrm{N}\}$

   $\{w : \#_a(w) = \#_b(w)\}$

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathsf{N}\}$

   $\{w : \#_a(w) = \#_b(w)\}$

   $\{w : k_1\#_a(w) = k_2\#_b(w)\}$

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathrm{N}\}$

   $\{w : \#_a(w) = \#_b(w)\}$

   $\{w : k_1\#_a(w) = k_2\#_b(w)\}$

   $\{a^n\}$ (Interesting: Small CFL, Large NFA)

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathrm{N}\}$

   $\{w : \#_a(w) = \#_b(w)\}$

   $\{w : k_1\#_a(w) = k_2\#_b(w)\}$

   $\{a^n\}$ (Interesting: Small CFL, Large NFA)

2. Chomsky Normal Form. Needed to make size comparisons.

What We Have Covered: Context Free Languages

1. **Examples of CFL's**

   $\{a^{k_1 n} b^{k_2 n} : n \in \mathrm{N}\}$

   $\{w : \#_a(w) = \#_b(w)\}$

   $\{w : k_1 \#_a(w) = k_2 \#_b(w)\}$

   $\{a^n\}$ (Interesting: Small CFL, Large NFA)
2. Chomsky Normal Form. Needed to make size comparisons.
3. Closure Properties.

What We Have Covered: Context Free Languages

1. **Examples of CFL's**
   $\{a^{k_1 n} b^{k_2 n} : n \in \mathsf{N}\}$
   $\{w : \#_a(w) = \#_b(w)\}$
   $\{w : k_1 \#_a(w) = k_2 \#_b(w)\}$
   $\{a^n\}$ (Interesting: Small CFL, Large NFA)
2. Chomsky Normal Form. Needed to make size comparisons.
3. Closure Properties.
4. Non-CFL's:
   If $L \subseteq a^*$ and not regular, than not CFL.
   If need to keep track of TWO things then NOT CFL.
   E.g., $\{a^n b^n c^n : n \in \mathsf{N}\}$

Equivalence of DFA, NFA, REGEX

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX:

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX: $R(i, j, k)$ Dynamic Programming. $|\alpha|$ is exp in number of states.

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX: $R(i,j,k)$ Dynamic Programming. $|\alpha|$ is exp in number of states.
2. L REGEX $\rightarrow$ L NFA:

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX: $R(i,j,k)$ Dynamic Programming. $|\alpha|$ is exp in number of states.
2. L REGEX $\rightarrow$ L NFA: induction on formation of a REGEX.

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX: $R(i,j,k)$ Dynamic Programming. $|\alpha|$ is exp in number of states.
2. L REGEX $\rightarrow$ L NFA: induction on formation of a REGEX.
3. L NFA $\rightarrow$ L DFA:

Equivalence of DFA, NFA, REGEX

1. L DFA $\rightarrow$ L REGEX: $R(i,j,k)$ Dynamic Programming. $|\alpha|$ is exp in number of states.
2. L REGEX $\rightarrow$ L NFA: induction on formation of a REGEX.
3. L NFA $\rightarrow$ L DFA: powerset construction. States blowup exponentially.

Closure Properties

1. **Union** What to use?

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?
   DFA: Swap final and non-final states.

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?
   DFA: Swap final and non-final states.
4. **Concatenation** What to use?

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?
   DFA: Swap final and non-final states.
4. **Concatenation** What to use?
   NFA: e-transition from one machine to the other.
   REGEX: By Def.

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: e-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?
   DFA: Swap final and non-final states.
4. **Concatenation** What to use?
   NFA: e-transition from one machine to the other.
   REGEX: By Def.
5. **Star** What to use?

Closure Properties

1. **Union** What to use?
   DFA: Cross Product Construction, or
   REGEX: by definition, or
   NFA: *e*-transitions.
2. **Intersection** What to use?
   DFA: Cross Product Construction.
   NFA: Cross Product Construction.
3. **Complimentation** What to use?
   DFA: Swap final and non-final states.
4. **Concatenation** What to use?
   NFA: *e*-transition from one machine to the other.
   REGEX: By Def.
5. **Star** What to use?
   NFA: transitions from final to new start/final to start.
   REGEX: By Def.

Pumping Lemma

Pumping Lemma If L is regular then there exists n_0 and n_1 such that the following holds:
For all $w \in L$, $|w| \geq n_0$ there exists x, y, z such that:

Pumping Lemma

Pumping Lemma If L is regular then there exists n_0 and n_1 such that the following holds:
For all $w \in L$, $|w| \geq n_0$ there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.

Pumping Lemma

Pumping Lemma If L is regular then there exists n_0 and n_1 such that the following holds:
For all $w \in L$, $|w| \geq n_0$ there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.
2. $|xy| \leq n_1$.

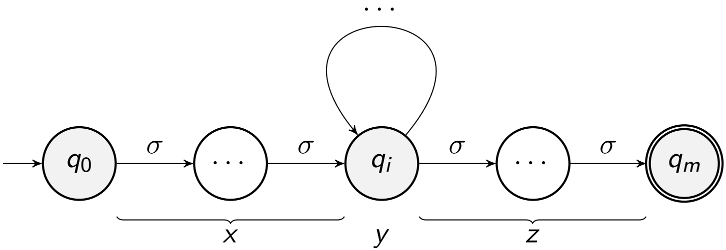
Pumping Lemma

Pumping Lemma If L is regular then there exists n_0 and n_1 such that the following holds:
For all $w \in L$, $|w| \geq n_0$ there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.
2. $|xy| \leq n_1$.
3. For all $i \geq 0$, $xy^iz \in L$.

Proof is picture on the next slide.

Proof by Pictures



How We Use the Pumping Lemma (PL)

We restate it in the way that we use it.

Pumping Lemma If L is reg then **for large enough strings w in *L*** there exists x, y, z such that:

How We Use the Pumping Lemma (PL)

We restate it in the way that we use it.

Pumping Lemma If L is reg then **for large enough strings w in *L*** there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.

How We Use the Pumping Lemma (PL)

We restate it in the way that we use it.

Pumping Lemma If L is reg then **for large enough strings w in *L*** there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.
2. $|xy|$ **is short** .

How We Use the Pumping Lemma (PL)

We restate it in the way that we use it.

Pumping Lemma If L is reg then **for large enough strings w in *L*** there exists x, y, z such that:

1. $w = xyz$ and $y \neq e$.
2. $|xy|$ **is short** .
3. for all i, $xy^iz \in L$.

We then find some i such that $xy^iz \notin L$ for the contradiction.

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

Take w long enough so that the xy part only has a's.

$L_1 = \{a^n b^n : n \in \mathbb{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

Take w long enough so that the xy part only has a's.
$x = a^{m_1}$, $y = a^{m_2}$, $z = a^{n-m_1-m_2} b^n$. Note $m_2 \geq 1$.

$L_1 = \{a^n b^n : n \in \mathbb{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

Take w long enough so that the xy part only has a's.
$x = a^{m_1}$, $y = a^{m_2}$, $z = a^{n-m_1-m_2} b^n$. Note $m_2 \geq 1$.
Take $i = 2$ to get

$$a^{m_1} a^{m_2} a^{m_2} a^{n-m_1-m_2} b^n \in L_1$$

$L_1 = \{a^n b^n : n \in \mathbf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

Take w long enough so that the xy part only has a's.
$x = a^{m_1}$, $y = a^{m_2}$, $z = a^{n-m_1-m_2} b^n$. Note $m_2 \geq 1$.
Take $i = 2$ to get

$$a^{m_1} a^{m_2} a^{m_2} a^{n-m_1-m_2} b^n \in L_1$$

$$a^{n+m_2} b^n \in L_1$$

$L_1 = \{a^n b^n : n \in \mathsf{N}\}$ is Not Regular

Assume L_1 reg. by PL for long enough string $a^n b^n \in L_1$ there exists x, y, z such that:

1. $y \neq e$.
2. $|xy|$ is short.
3. For all $i \geq 0$, $xy^i z \in L_1$.

Take w long enough so that the xy part only has a's.
$x = a^{m_1}$, $y = a^{m_2}$, $z = a^{n-m_1-m_2} b^n$. Note $m_2 \geq 1$.
Take $i = 2$ to get

$$a^{m_1} a^{m_2} a^{m_2} a^{n-m_1-m_2} b^n \in L_1$$

$$a^{n+m_2} b^n \in L_1$$

Contradiction since $m_2 \geq 1$.

$L_2 = \{w : \sharp a(w) = \sharp b(w)\}$ is Not Regular

Proof: Same Proof as L_1 not Reg : Still look at $a^m b^m$.
Key Pumping Lemma says for ALL long enough $w \in L$.

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular

$L_4 = \{a^{n^2} : n \in \mathrm{N}\}$ is Not Regular

Proof

By Pumping Lemma for long enough $a^{n^2} \in L_4$ there exists $x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n_3}$ such that

$$a^{n_1}(a^{n_2})^i a^{n_3} \in L_4$$

$L_4 = \{a^{n^2} : n \in \mathrm{N}\}$ is Not Regular

Proof

By Pumping Lemma for long enough $a^{n^2} \in L_4$ there exists $x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n_3}$ such that

$$a^{n_1}(a^{n_2})^i a^{n_3} \in L_4$$

$$(\forall i \geq 0)[n_1 + in_2 + n_3 \text{ is a square}].$$

$L_4 = \{a^{n^2} : n \in \mathrm{N}\}$ is Not Regular

Proof

By Pumping Lemma for long enough $a^{n^2} \in L_4$ there exists $x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n_3}$ such that

$$a^{n_1}(a^{n_2})^i a^{n_3} \in L_4$$

$$(\forall i \geq 0)[n_1 + in_2 + n_3 \text{ is a square}].$$

$$(n_1 + n_3) = x^2$$

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular

Proof

By Pumping Lemma for long enough $a^{n^2} \in L_4$ there exists $x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n_3}$ such that

$$a^{n_1}(a^{n_2})^i a^{n_3} \in L_4$$

$$(\forall i \geq 0)[n_1 + in_2 + n_3 \text{ is a square}].$$

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x + 1)^2$$

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular

Proof

By Pumping Lemma for long enough $a^{n^2} \in L_4$ there exists $x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n_3}$ such that

$$a^{n_1}(a^{n_2})^i a^{n_3} \in L_4$$

$$(\forall i \geq 0)[n_1 + in_2 + n_3 \text{ is a square}].$$

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x+1)^2$$

$$(n_1 + n_3) + 2n_2 \geq (x+2)^2$$

$L_4 = \{a^{n^2} : n \in \mathbf{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x+1)^2$$

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x + 1)^2$$

$$(n_1 + n_3) + 2n_2 \geq (x + 2)^2$$

$L_4 = \{a^{n^2} : n \in \mathbf{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x+1)^2$$

$$(n_1 + n_3) + 2n_2 \geq (x+2)^2$$

$$(n_1 + n_3) + in_2 \geq x^2 + 2ix + i^2$$

$L_4 = \{a^{n^2} : n \in \mathbf{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x + 1)^2$$

$$(n_1 + n_3) + 2n_2 \geq (x + 2)^2$$

$$(n_1 + n_3) + in_2 \geq x^2 + 2ix + i^2$$

$$(n_1 + n_3) + in_2 \geq i^2$$

$L_4 = \{a^{n^2} : n \in \mathbb{N}\}$ is Not Regular (cont)

$$(n_1 + n_3) = x^2$$

$$(n_1 + n_3) + n_2 \geq (x+1)^2$$

$$(n_1 + n_3) + 2n_2 \geq (x+2)^2$$

$$(n_1 + n_3) + in_2 \geq x^2 + 2ix + i^2$$

$$(n_1 + n_3) + in_2 \geq i^2$$

$$\frac{(n_1 + n_3)}{i} + n_2 \geq i$$

As i increases the LHS decreases and the RHS goes to infinity, so this cannot hold for all i.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

So what to do? Let's go back to the pumping lemma with a carefully chosen string.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

So what to do? Let's go back to the pumping lemma with a carefully chosen string.

$w = a^n b^{n-1} c^n$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

So what to do? Let's go back to the pumping lemma with a carefully chosen string.

$w = a^n b^{n-1} c^n$.

$x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n-n_1-n_2} b^{n-1} c^n$.

$L_8 = \{a^{n_1}b^mc^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

So what to do? Let's go back to the pumping lemma with a carefully chosen string.

$w = a^n b^{n-1} c^n$.

$x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n-n_1-n_2} b^{n-1} c^n$.

For all $i \geq 0$, $xy^i z \in L_8$.

$L_8 = \{a^{n_1}b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Problematic Neither pumping on the left or on the right works. (I give proof that uses $i = 0$ case. Students came up with two other proofs. (1) Use closure of REG under PREFIX, (2) Carefully pump in the middle-not safe for work.

So what to do? Let's go back to the pumping lemma with a carefully chosen string.

$w = a^n b^{n-1} c^n$.

$x = a^{n_1}$, $y = a^{n_2}$, $z = a^{n-n_1-n_2} b^{n-1} c^n$.

For all $i \geq 0$, $xy^i z \in L_8$.

$$xy^i z = a^{n_1 + in_2 + (n-n_1-n_2)} b^{n-1} c^n$$

$L_8 = \{a^{n_1}b^{m}c^{n_2} : n_1, n_2 > m\}$ is Not Regular (Cont)

$$xy^iz = a^{n_1+in_2+(n-n_1-n_2)}b^{n-1}c^n$$

$L_8 = \{a^{n_1}b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular (Cont)

$$xy^iz = a^{n_1+in_2+(n-n_1-n_2)}b^{n-1}c^n$$

For all i $xy^iz = a^{n_1+in_2+(n-n_1-n_2)}b^{n-1}c^n \in L_8$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular (Cont)

$$xy^i z = a^{n_1 + in_2 + (n - n_1 - n_2)} b^{n-1} c^n$$

For all i $xy^i z = a^{n_1 + in_2 + (n - n_1 - n_2)} b^{n-1} c^n \in L_8$.

Key We are used to thinking of i large. But we can also take $i = 0$, cut out that part of the word. We take $i = 0$ to get

$$xy^0 z = a^{n - n_2} b^{n-1} c^n$$

$L_8 = \{a^{n_1}b^mc^{n_2} : n_1, n_2 > m\}$ is Not Regular (Cont)

$$xy^iz = a^{n_1+in_2+(n-n_1-n_2)}b^{n-1}c^n$$

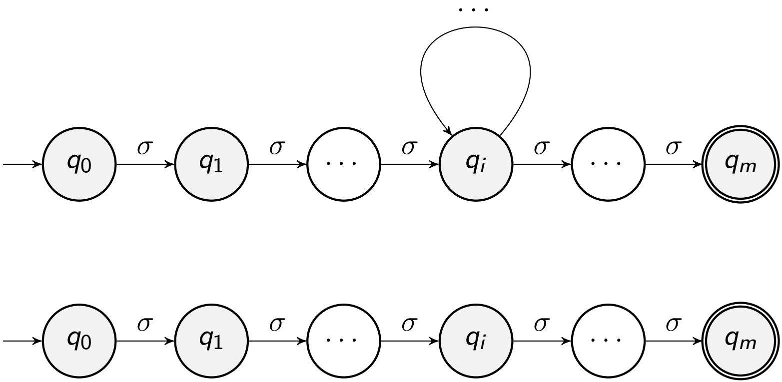
For all i $xy^iz = a^{n_1+in_2+(n-n_1-n_2)}b^{n-1}c^n \in L_8$.

Key We are used to thinking of i large. But we can also take $i = 0$, cut out that part of the word. We take $i = 0$ to get

$$xy^0z = a^{n-n_2}b^{n-1}c^n$$

Since $n_2 \ge 1$, we have that $\sharp a(xy^0z) < n \le n - 1 = \sharp b(xy^0z)$.
Hence $xy^0z \notin L_8$.
(There were two other proofs by students: One used that REG closed under PREFIX, and one managed to pump in the middle.)

$i = 0$ Case as a Picture



Answer to SUBSEQ Problem: CFL

If L is CFL than $SUBSEQ(L)$ is CFL.

Answer to SUBSEQ Problem: CFL

If L is CFL than $SUBSEQ(L)$ is CFL. YES.

Answer to SUBSEQ Problem: CFL

If L is CFL than $SUBSEQ(L)$ is CFL. YES.
Let M be a CFL for L in Chomsky Normal Form.

We form a CFL $SUBSEQ(L)$.

For every rule $A \rightarrow \sigma$ we add $A \rightarrow \epsilon$.

Context Free Languages

Definition

A **Context Free Grammar (CFL)** is (V, Σ, P, S)

- V is set of **nonterminals**
- Σ is the **alphabet** , also called **terminals**
- $P \subseteq V \times (V \cup \Sigma)^*$ are the **productions** or **rules**
- $S \in V$ is the start symbol.

Context Free Languages

Definition

A **Context Free Grammar (CFL)** is (V, Σ, P, S)

- V is set of **nonterminals**
- Σ is the **alphabet** , also called **terminals**
- $P \subseteq V \times (V \cup \Sigma)^*$ are the **productions** or **rules**
- $S \in V$ is the start symbol.

$L(G)$ is the set of strings generated by CFL G.
A **Context Free Lang (CFL)** is a lang that is $L(G)$ for some CFL G.

Context Free Languages

Definition

A **Context Free Grammar (CFL)** is (V, Σ, P, S)

- V is set of **nonterminals**
- Σ is the **alphabet** , also called **terminals**
- $P \subseteq V \times (V \cup \Sigma)^*$ are the **productions** or **rules**
- $S \in V$ is the start symbol.

$L(G)$ is the set of strings generated by CFL G.
A **Context Free Lang (CFL)** is a lang that is $L(G)$ for some CFL G.

A CFL is in **Chomsky Normal Form CNF)** if all of he productions are either of the form
$A \rightarrow BC$
$A \rightarrow \sigma$ where $\sigma \in \Sigma$
$A \rightarrow e$ (I didn't include it in class, but I am now.)
Note: If G is a CFL hen there exists a CNF CFL that generates it.

Examples of CFL's that are NOT Regular

$\{a^n b^n : n \in \mathrm{N}\}$
$S \rightarrow aSb|e$

Examples of CFL's that are NOT Regular

$\{a^n b^n : n \in \mathrm{N}\}$
$S \rightarrow aSb|e$

$\{w : \#_a(w) = \#_b(w)\}$
$S \rightarrow aSbS$
$S \rightarrow bSaS$
$S \rightarrow SS$
$S \rightarrow e$
To prove it works requires a proof by induction

Examples of CFL's that are NOT Regular

$\{a^n b^n : n \in \mathrm{N}\}$
$S \to aSb|e$

$\{w : \#_a(w) = \#_b(w)\}$
$S \to aSbS$
$S \to bSaS$
$S \to SS$
$S \to e$
To prove it works requires a proof by induction
Not to worry, I will ASSUME you could do such a proof and hence WILL NOT make you.

Examples of Langs with Small CFL's, Large NFA's

$L = \{a^n\}$

- NFA **requires** $\geq n - 2$ states. Lets prove it

Examples of Langs with Small CFL's, Large NFA's

$L = \{a^n\}$

- NFA **requires** $\geq n-2$ states. Lets prove it
  If M is an NFA with $\leq n-2$ states then find a path from the start state to the final state. Let a^m be the shortest string that take you from the start state to the final state. Since the number of states is $\leq n-2$, $m \leq n-2$. So we have a^m accepted when it should not be. Contradiction.
- There is a CNF CFL with $\leq 2\log_2 n$ rules.
  For $n = 2^n$ VERY EASY. If not then have to write n as a sum of powers of 2. Example on next slide.

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$
$S \to XY$

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$
$S \to XY$ We make $X \Rightarrow a^8$ and $Y \Rightarrow a^2$.

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$
$S \rightarrow XY$ We make $X \Rightarrow a^8$ and $Y \Rightarrow a^2$.
$X \rightarrow X_1X_1$
$X_1 \rightarrow X_2X_2$
$X_2 \rightarrow X_3X_3$
$X_3 \rightarrow a$
$Y \rightarrow Y_1Y_1$
$Y_1 \rightarrow a$

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$

$S \rightarrow XY$ We make $X \Rightarrow a^8$ and $Y \Rightarrow a^2$.

$X \rightarrow X_1X_1$

$X_1 \rightarrow X_2X_2$

$X_2 \rightarrow X_3X_3$

$X_3 \rightarrow a$

$Y \rightarrow Y_1Y_1$

$Y_1 \rightarrow a$

Can shorten a bit: We need $Y \Rightarrow aa$, so can just use X_2.

CNF CFG for $\{a^{10}\}$

$10 = 2^3 + 2^1$
$S \to XY$ We make $X \Rightarrow a^8$ and $Y \Rightarrow a^2$.
$X \to X_1X_1$
$X_1 \to X_2X_2$
$X_2 \to X_3X_3$
$X_3 \to a$
$Y \to Y_1Y_1$
$Y_1 \to a$
Can shorten a bit: We need $Y \Rightarrow aa$, so can just use X_2.
$S \to XX_2$
$X \to X_1X_1$
$X_1 \to X_2X_2$
$X_2 \to X_3X_3$
$X_3 \to a$