BILL AND NATHAN START RECORDING

Context Free Languages

Why Are Context Free Languages Important

I am supposed to say:

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:
1) Languages that are CFL but not Regular.

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:
1) Languages that are CFL but not Regular.
2) CFL's are all in P (poly time).

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:
1) Languages that are CFL but not Regular.
2) CFL's are all in P (poly time).
3) Closure properties of CFLs.

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:

1) Languages that are CFL but not Regular.
2) CFL's are all in P (poly time).
3) Closure properties of CFLs.
4) Languages that require a LARGE NFA but a SMALL CFG.

Why Are Context Free Languages Important

I am supposed to say:

Most prog langs are Context Free Languages

However, this is not quite true. PL people - discuss!

However, most programming languages are **almost context free.**

Our interest in CFL's is:

1) Languages that are CFL but not Regular.
2) CFL's are all in P (poly time).
3) Closure properties of CFLs.
4) Languages that require a LARGE NFA but a SMALL CFG.
5) Which languages are **not** context free?

Examples of Context Free Grammars

$S \to aSb$
$S \to e$

The set of all strings **Generated** is

Examples of Context Free Grammars

$S \to aSb$
$S \to e$

The set of all strings **Generated** is

$$L = \{a^n b^n : n \in \mathrm{N}\}$$

Examples of Context Free Grammars

$S \rightarrow aSb$
$S \rightarrow e$

The set of all strings **Generated** is

$$L = \{a^n b^n : n \in \mathrm{N}\}$$

Note L is a context free lang that is not regular.

Context Free Grammar for $\{a^{2^n} b^n : n \in \mathbf{N}\}$

$S \rightarrow aaSb$
$S \rightarrow e$

The set of all strings **Generated** is

Context Free Grammar for $\{a^{2n}b^n : n \in \mathbb{N}\}$

$S \rightarrow aaSb$
$S \rightarrow e$

The set of all strings **Generated** is

$$L = \{a^{2n}b^n : n \in \mathrm{N}\}$$

Context Free Grammar for $\{a^{2n}b^n : n \in \mathbb{N}\}$

$S \to aaSb$
$S \to e$

The set of all strings **Generated** is

$$L = \{a^{2n}b^n : n \in \mathbb{N}\}$$

Note L is context free lang that is not regular.

Context Free Grammar for $\{a^m b^n : m > n\}$

DO WITH YOUR TABLE MATES- NO PERSON IS AN ISLAND.

Context Free Grammar for $\{a^m b^n : m > n\}$

DO WITH YOUR TABLE MATES- NO PERSON IS AN ISLAND.

$S \rightarrow AT$

$T \rightarrow aTb$

$T \rightarrow e$

$A \rightarrow Aa$

$A \rightarrow a$

Context Free Grammars

Def A **Context Free Grammar** is a tuple $G = (N, \Sigma, R, S)$

- N is a finite set of **nonterminals**.
- Σ is a finite **alphabet**. Note $\Sigma \cap N = \emptyset$.
- $R \subseteq N \times (N \cup \Sigma)^*$ and are called **Rules**.
- $S \in N$, the **start symbol**.

L(G)

If A is non-terminal then the CFG gives us gives us rules like:

- $A \rightarrow AB$
- $A \rightarrow a$

L(G)

If A is non-terminal then the CFG gives us gives us rules like:

- $A \rightarrow AB$
- $A \rightarrow a$

For any string of **terminals and non-terminals** α, $A \Rightarrow \alpha$ means that, starting from A, some combination of the rules produces α.

L(G)

If A is non-terminal then the CFG gives us gives us rules like:

- $A \to AB$
- $A \to a$

For any string of **terminals and non-terminals** α, $A \Rightarrow \alpha$ means that, starting from A, some combination of the rules produces α.

Examples:

- $A \Rightarrow a$
- $A \Rightarrow aB$

L(G)

If A is non-terminal then the CFG gives us gives us rules like:

- $A \rightarrow AB$
- $A \rightarrow a$

For any string of **terminals and non-terminals** α, $A \Rightarrow \alpha$ means that, starting from A, some combination of the rules produces α.

Examples:

- $A \Rightarrow a$
- $A \Rightarrow aB$

Then, if w is string of **terminals only**, we define $L(G)$ by:

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow w\}$$

Number of a's = Number of b's

Is

$$L = \{w \mid \#_a(w) = \#_b(w)\}$$

context free?

YES

Let G be the CFG

$S \to aSb$

$S \to bSa$

$S \to SS$

$S \to e$

YES

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

YES

Let G be the CFG

$S \to aSb$

$S \to bSa$

$S \to SS$

$S \to e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

YES

Let G be the CFG

$S \to aSb$

$S \to bSa$

$S \to SS$

$S \to e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Contrast

YES

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Contrast

Never proved a DFA recognized language we claimed it did.

YES

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Contrast

Never proved a DFA recognized language we claimed it did.

Never proved a regex generated the language we claimed it did.

YES

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Contrast

Never proved a DFA recognized language we claimed it did.

Never proved a regex generated the language we claimed it did.

Gasarch's Principle Never prove an obvious Theorem.

YES

Let G be the CFG
$S \rightarrow aSb$
$S \rightarrow bSa$
$S \rightarrow SS$
$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Contrast
Never proved a DFA recognized language we claimed it did.
Never proved a regex generated the language we claimed it did.
Gasarch's Principle Never prove an obvious Theorem.
(Exception: a course on foundations. I proved $x + y = y + x$.)

Deserves a Proof But...

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Deserves a Proof But...

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Deserves a Proof But...

Let G be the CFG

$S \rightarrow aSb$

$S \rightarrow bSa$

$S \rightarrow SS$

$S \rightarrow e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Deserves a Proof But...

Let G be the CFG

$S \to aSb$

$S \to bSa$

$S \to SS$

$S \to e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Note Proof is messy.

Deserves a Proof But. . .

Let G be the CFG

$S \to aSb$

$S \to bSa$

$S \to SS$

$S \to e$

Thm $L(G) = \{w \mid \#_a(w) = \#_b(w)\}$.

Note This Theorem is **not obvious**. Deserves a proof!

Note Proof is messy.

Solution The proof is on the slides, but I won't go over it, and you don't need to know it for a HW or Exam.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \to aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.

Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.

Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.

Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.

Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.

Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

Ind Hyp If $S \Rightarrow \beta$ in $n-1$ steps then $\#_a(\beta) = \#_b(\beta)$.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG
$S \to aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.
Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

Ind Hyp If $S \Rightarrow \beta$ in $n - 1$ steps then $\#_a(\beta) = \#_b(\beta)$.

Ind Step Assume $S \Rightarrow \alpha$ in n steps. Look at the last step.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.
Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

Ind Hyp If $S \Rightarrow \beta$ in $n-1$ steps then $\#_a(\beta) = \#_b(\beta)$.

Ind Step Assume $S \Rightarrow \alpha$ in n steps. Look at the last step.
Case 1 $S \Rightarrow \alpha' S \alpha'' \rightarrow \alpha' aSb\alpha$. By IH $\#_a(\alpha' S \alpha'') = \#_b(\alpha' S \alpha'')$.
$\#_a(\alpha' aSb\alpha'') = \#_b(\alpha' S \alpha'') + 1$.
$\#_b(\alpha' aSb\alpha'') = \#_b(\alpha' S \alpha'') + 1$.

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.
Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

Ind Hyp If $S \Rightarrow \beta$ in $n-1$ steps then $\#_a(\beta) = \#_b(\beta)$.

Ind Step Assume $S \Rightarrow \alpha$ in n steps. Look at the last step.
Case 1 $S \Rightarrow \alpha' S \alpha'' \rightarrow \alpha' aSb\alpha$. By IH $\#_a(\alpha' S\alpha'') = \#_b(\alpha' S\alpha'')$.
$\#_a(\alpha' aSb\alpha'') = \#_b(\alpha' S\alpha'') + 1$.
$\#_b(\alpha' aSb\alpha'') = \#_b(\alpha' S\alpha'') + 1$.

Hence
$\#_a(\alpha' aSb\alpha'') = \#_b(\alpha' aSb\alpha'')$

$L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $L(G) \subseteq \{w \mid \#_a(w) = \#_b(w)\}$. We prove something stronger.
Let $L(G)' = \{\alpha \in \{S, a, b\}^* : S \Rightarrow \alpha\}$ (Note that we allow S in α.)

Thm $L(G)' \subseteq \{w \mid \#_a(w) = \#_b(w)\}$.

This is by induction on numb of steps in the derivation from S.

Base Case In one step can only get $\alpha \in \{aSb, bSa, SS, e\}$.

Ind Hyp If $S \Rightarrow \beta$ in $n-1$ steps then $\#_a(\beta) = \#_b(\beta)$.

Ind Step Assume $S \Rightarrow \alpha$ in n steps. Look at the last step.
Case 1 $S \Rightarrow \alpha' S \alpha'' \rightarrow \alpha' aSb\alpha$. By IH $\#_a(\alpha' S \alpha'') = \#_b(\alpha' S \alpha'')$.
$\#_a(\alpha' aSb\alpha'') = \#_b(\alpha' S \alpha'') + 1$.
$\#_b(\alpha' aSb\alpha'') = \#_b(\alpha' S \alpha'') + 1$.

Hence
$\#_a(\alpha' aSb\alpha'') = \#_b(\alpha' aSb\alpha'')$

Case 2 Other cases for last step similar.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG

$S \to aSb \mid bSa \mid SS \mid e$

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \to aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \to aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \to aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \to e$.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \leq n - 1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \leq n - 1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

Ind Step Let w be such that $\#_a(w) = \#_b(w)$.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \leq n - 1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

Ind Step Let w be such that $\#_a(w) = \#_b(w)$.
Case 1 $w = aw'b$. Then $w' \in L(G)$. By IH $S \Rightarrow w'$.
$S \rightarrow aSb \Rightarrow aw'b$.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \leq n-1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

Ind Step Let w be such that $\#_a(w) = \#_b(w)$.
Case 1 $w = aw'b$. Then $w' \in L(G)$. By IH $S \Rightarrow w'$.
$S \rightarrow aSb \Rightarrow aw'b$.

Case 2 $w = bw'a$. Similar.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \le n - 1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

Ind Step Let w be such that $\#_a(w) = \#_b(w)$.
Case 1 $w = aw'b$. Then $w' \in L(G)$. By IH $S \Rightarrow w'$.
$S \rightarrow aSb \Rightarrow aw'b$.

Case 2 $w = bw'a$. Similar.

Case 3 $w = aw'a$. This is first NON-OBVIOUS part!

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Thm $\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$.
This is not obvious!

We must show that **every** w with $\#_a(w) = \#_b(w)$ can be generated.
DISCUSS WITH YOUR TABLE MATES.
We use induction on $|w|$.

Base Case $|w| = 0$. So $w = e$. Can be generated by $S \rightarrow e$.

Ind Hyp If $|w'| \leq n - 1$ and $\#_a(w') = \#_b(w')$ then $w' \in L(G)$.

Ind Step Let w be such that $\#_a(w) = \#_b(w)$.
Case 1 $w = aw'b$. Then $w' \in L(G)$. By IH $S \Rightarrow w'$.
$S \rightarrow aSb \Rightarrow aw'b$.

Case 2 $w = bw'a$. Similar.

Case 3 $w = aw'a$. This is first NON-OBVIOUS part! Next Slide.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG

$S \rightarrow aSb \mid bSa \mid SS \mid e$

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:
a: $\#_a(a) > \#_b(a)$

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \to aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:
a: $\#_a(a) > \#_b(a)$

For all $2 \le i \le n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
But NOT both.

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:
a: $\#_a(a) > \#_b(a)$

For all $2 \le i \le n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
But NOT both.

$\#_a(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2} - 1$
$\#_b(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2}$

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:
a: $\#_a(a) > \#_b(a)$

For all $2 \leq i \leq n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
But NOT both.

$\#_a(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2} - 1$
$\#_b(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2}$

Hence

$\{w \mid \#_a(w) = \#_b(w)\} \subseteq L(G)$

Let G be the CFG
$S \rightarrow aSb \mid bSa \mid SS \mid e$

Case 3 $w = aw'a$. Let $w = a\sigma_2 \cdots \sigma_{n-1}a$. Look at prefixes of w:
a: $\#_a(a) > \#_b(a)$

For all $2 \le i \le n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
But NOT both.

$\#_a(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2} - 1$
$\#_b(a\sigma_2 \cdots \sigma_{n-1}) = \frac{n}{2}$

Hence

$\#_a(a\sigma_2 \cdots \sigma_{n-1}) < \#_b(a\sigma_2 \cdots \sigma_{n-1})$

Recap

1) a: $\#_a(a) > \#_b(a)$

Recap

1) a: $\#_a(a) > \#_b(a)$

2) For all $2 \le i \le n-1$, EITHER

$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1.$

OR

$\#_b(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1.$

Recap

1) a: $\#_a(a) > \#_b(a)$

2) For all $2 \leq i \leq n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.

3) $\#_a(a\sigma_2 \cdots \sigma_{n-1}) < \#_b(a\sigma_2 \cdots \sigma_{n-1})$
Hence there exists $2 \leq i \leq n-1$
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1})$.

Recap

1) a: $\#_a(a) > \#_b(a)$

2) For all $2 \leq i \leq n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.

3) $\#_a(a\sigma_2 \cdots \sigma_{n-1}) < \#_b(a\sigma_2 \cdots \sigma_{n-1})$

Hence there exists $2 \leq i \leq n-1$
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1})$.

So $w = w'w''$ where $w, w' \in L(G)$. Since $|w'| < |w|$ and
$|w''| < |w|$, by IH
$S \Rightarrow w'$ and $S \Rightarrow w''$.

Recap

1) a: $\#_a(a) > \#_b(a)$

2) For all $2 \leq i \leq n-1$, EITHER
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.
OR
$\#_b(a\sigma_2 \cdots \sigma_i) = \#_a(a\sigma_2 \cdots \sigma_{i-1}) + 1$.

3) $\#_a(a\sigma_2 \cdots \sigma_{n-1}) < \#_b(a\sigma_2 \cdots \sigma_{n-1})$
Hence there exists $2 \leq i \leq n-1$
$\#_a(a\sigma_2 \cdots \sigma_i) = \#_b(a\sigma_2 \cdots \sigma_{i-1})$.
So $w = w'w''$ where $w, w' \in L(G)$. Since $|w'| < |w|$ and $|w''| < |w|$, by IH
$S \Rightarrow w'$ and $S \Rightarrow w''$.

So
$S \rightarrow SS \Rightarrow w'w'' = w$.

Example of a Lang that is NOT a CFL

1) $\{a^n b^n c^n : n \in \mathrm{N}\}$ is NOT a CFL.

Example of a Lang that is NOT a CFL

1) $\{a^n b^n c^n : n \in N\}$ is NOT a CFL.

2) $\{a^{n^2} : n \in N\}$ is NOT a CFL.

Example of a Lang that is NOT a CFL

1) $\{a^n b^n c^n : n \in \mathrm{N}\}$ is NOT a CFL.

2) $\{a^{n^2} : n \in \mathrm{N}\}$ is NOT a CFL.

3) If $L \subseteq a^*$ and L is not regular than L is not a CFL.

Example of a Lang that is NOT a CFL

1) $\{a^n b^n c^n : n \in \mathrm{N}\}$ is NOT a CFL.

2) $\{a^{n^2} : n \in \mathrm{N}\}$ is NOT a CFL.

3) If $L \subseteq a^*$ and L is not regular than L is not a CFL.

We will not be proving Langs NOT CFL.

CLOSURE PROPERTIES AND REG$\subset$ CFL

Closure Properties: PROVE or DISPROVE

If L_1, L_2 are Context Free Languages then

1. IS $L_1 \cup L_2$ is a context free Lang?
2. IS $L_1 \cap L_2$ is a context free Lang?
3. IS $L_1 \cdot L_2$ is a context free Lang?
4. IS $\overline{L_1}$ is a context free Lang?
5. IS L_1^* is a context free Lang?

DISCUSS WITH YOUR TABLE MATES.

L_1, L_2 CFL $\rightarrow L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

L_1, L_2 CFL $\rightarrow$ $L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cup L_2$.
$L_1 \cup L_2$ is CFL via CFG (N, Σ, R, S) where

L_1, L_2 CFL $\rightarrow$ $L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cup L_2$.
$L_1 \cup L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

L_1, L_2 CFL $\rightarrow$ $L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cup L_2$.
$L_1 \cup L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is start state.

L_1, L_2 CFL $\rightarrow L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cup L_2$.
$L_1 \cup L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is start state.

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 \mid S_2\}$

L_1, L_2 CFL $\rightarrow$ $L_1 \cup L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cup L_2$.
$L_1 \cup L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is start state.

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 \mid S_2\}$

Note We assume $N_1 \cap N_2 = \emptyset$.

Finite vs Infinite Union

If L_1 and L_2 are regular then $L_1 \cup L_2$ is regular.

This is true for 3 languages or 4 languages or 98 languages.

Finite vs Infinite Union

If L_1 and L_2 are regular then $L_1 \cup L_2$ is regular.

This is true for 3 languages or 4 languages or 98 languages.

But if $L_1, L_2, L_3, \cdots$ is an **infinite** set of regular languages, is $L_1 \cup L_2 \cup ...$ regular?

Finite vs Infinite Union

If L_1 and L_2 are regular then $L_1 \cup L_2$ is regular.

This is true for 3 languages or 4 languages or 98 languages.

But if $L_1, L_2, L_3, \cdots$ is an **infinite** set of regular languages, is $L_1 \cup L_2 \cup ...$ regular?

No, because:

- $L_1 = \{ab\}$ is regular.
- $L_k = \{a^k b^k\}$ is regular.
- $L_1 \cup L_2 \cup \cdots = \{a^n b^n : n \in \mathrm{N}\}$ is not regular.

Finite vs Infinite Union

If L_1 and L_2 are regular then $L_1 \cup L_2$ is regular.

This is true for 3 languages or 4 languages or 98 languages.

But if $L_1, L_2, L_3, \cdots$ is an **infinite** set of regular languages, is $L_1 \cup L_2 \cup \ldots$ regular?

No, because:

- $L_1 = \{ab\}$ is regular.
- $L_k = \{a^k b^k\}$ is regular.
- $L_1 \cup L_2 \cup \cdots = \{a^n b^n : n \in \mathrm{N}\}$ is not regular.

What about for CFLs?

- $L_1 = \{abc\}$ is a CFL.
- $L_k = \{a^k b^k c^k\}$ is a CFL.
- We will see later that $\bigcup_{i=1}^{\infty} L_i = \{a^n b^n c^n : n \in \mathrm{N}\}$ is not CFL.

L_1, L_2 CFL $\rightarrow L_1 \cap L_2$ CFL

NOT TRUE: $a^n b^n c^* \cap a^* b^n c^n = a^n b^n c^n$.

L_1, L_2 CFL $\rightarrow L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

L_1, L_2 CFL $\rightarrow L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cdot L_2$.
$L_1 \cdot L_2$ is CFL via CFG (N, Σ, R, S) where

L_1, L_2 CFL $\rightarrow L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cdot L_2$.
$L_1 \cdot L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

L_1, L_2 CFL $\rightarrow$ $L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cdot L_2$.
$L_1 \cdot L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is the start state.

L_1, L_2 CFL $\rightarrow L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cdot L_2$.
$L_1 \cdot L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is the start state.

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 \cdot S_2\}$.

L_1, L_2 CFL $\rightarrow L_1 \cdot L_2$ CFL

L_1 is CFL via CFG (N_1, Σ, R_1, S_1).
L_2 is CFL via CFG (N_2, Σ, R_2, S_2).

The following CFG generates $L_1 \cdot L_2$.
$L_1 \cdot L_2$ is CFL via CFG (N, Σ, R, S) where

$N = N_1 \cup N_2 \cup \{S\}$

S is the start state.

$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 \cdot S_2\}$.

Note We assume $N_1 \cap N_2 = \emptyset$.

L CFL $\rightarrow \overline{L}$ CFL

FALSE.

Let

$$L = \overline{\{a^n b^n c^n : n \in \mathrm{N}\}}$$

L CFL $\rightarrow \overline{L}$ CFL

FALSE.
Let

$$L = \overline{\{a^n b^n c^n : n \in \mathbb{N}\}}$$

This is a CFL. This will be a HW.

L CFL $\rightarrow$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

L CFL $\rightarrow$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where

L CFL $\rightarrow$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where
S' is new start nonterminal.

L CFL $\rightarrow$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where
S' is new start nonterminal.
$N' = N \cup \{S'\}$.

L CFL $\rightarrow$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where
S' is new start nonterminal.
$N' = N \cup \{S'\}$.
R' has R and also

L CFL $\to$ L^* CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where
S' is new start nonterminal.
$N' = N \cup \{S'\}$.
R' has R and also
$S' \to e$

L CFL $\rightarrow L^*$ CFL

L is CFL via CFG (N, Σ, R, S).

Here is a CFL for L^*: (N', Σ, R', S') where
S' is new start nonterminal.
$N' = N \cup \{S'\}$.
R' has R and also
$S' \rightarrow e$
$S' \rightarrow S'S$

REG contained in CFL

Thm If L is regular then L is CFL.
DISCUSS

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

Ind Hyp For all regex β with $|\beta| < n$ there exists CFG G such that $L(\beta) = L(G)$.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

Ind Hyp For all regex β with $|\beta| < n$ there exists CFG G such that $L(\beta) = L(G)$.

Ind Step $|\alpha| = n$.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

Ind Hyp For all regex β with $|\beta| < n$ there exists CFG G such that $L(\beta) = L(G)$.

Ind Step $|\alpha| = n$.
Case 1 $\alpha = \beta_1 \cup \beta_2$. By IH $L(\beta_1)$ and $L(\beta_2)$ are CFL's. By closure under $\cup$, $L(\alpha)$ is CFL.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

Ind Hyp For all regex β with $|\beta| < n$ there exists CFG G such that $L(\beta) = L(G)$.

Ind Step $|\alpha| = n$.
Case 1 $\alpha = \beta_1 \cup \beta_2$. By IH $L(\beta_1)$ and $L(\beta_2)$ are CFL's. By closure under $\cup$, $L(\alpha)$ is CFL.
Case 2 $\alpha = \beta_1 \cdot \beta_2$. By IH $L(\beta_1)$ and $L(\beta_2)$ are CFL's. By closure under $\cdot$, $L(\alpha)$ is CFL.

REG contained in CFL

For every **regex** α, $L(\alpha)$ is a CFL.

Prove by ind on the length of α.

Base Case $|\alpha| = 1$ then α is σ or e. Both $\{\sigma\}$ and $\{e\}$ are CFL's.

Ind Hyp For all regex β with $|\beta| < n$ there exists CFG G such that $L(\beta) = L(G)$.

Ind Step $|\alpha| = n$.
Case 1 $\alpha = \beta_1 \cup \beta_2$. By IH $L(\beta_1)$ and $L(\beta_2)$ are CFL's. By closure under $\cup$, $L(\alpha)$ is CFL.
Case 2 $\alpha = \beta_1 \cdot \beta_2$. By IH $L(\beta_1)$ and $L(\beta_2)$ are CFL's. By closure under $\cdot$, $L(\alpha)$ is CFL.
Case 3 $\alpha = \beta^*$. By IH $L(\beta)$ is CFL. By closure under $*$, $L(\alpha)$ is CFL.

Examples of CFL's and Size of CFG's

Size of CFGs

How big is a CFL for the language $\{aaaaaaaa\}$ (there are 8 a's).

Size of CFGs

How big is a CFL for the language $\{aaaaaaaa\}$ (there are 8 a's).

We could say the size is 1:

$$S \to aaaaaaaa$$

Size of CFGs

How big is a CFL for the language $\{aaaaaaaa\}$ (there are 8 a's).

We could say the size is 1:

$$S \rightarrow aaaaaaaa$$

This does not seem quite right.

Size of CFGs

How big is a CFL for the language $\{aaaaaaaa\}$ (there are 8 a's).

We could say the size is 1:

$$S \rightarrow aaaaaaaa$$

This does not seem quite right.

Next slide has a standard form for CFL's that make size make sense.

Chomsky Normal Form

Def CFG G is in **Chomsky Normal Form** if the rules are all of the following form:

Chomsky Normal Form

Def CFG G is in **Chomsky Normal Form** if the rules are all of the following form:
1) $A \rightarrow BC$ where $A, B, C \in N$ (nonterminals).

Chomsky Normal Form

Def CFG G is in **Chomsky Normal Form** if the rules are all of the following form:
1) $A \rightarrow BC$ where $A, B, C \in N$ (nonterminals).
2) $A \rightarrow \sigma$ (where $A \in N$ and $\sigma \in \Sigma$).

Chomsky Normal Form

Def CFG G is in **Chomsky Normal Form** if the rules are all of the following form:
1) $A \to BC$ where $A, B, C \in N$ (nonterminals).
2) $A \to \sigma$ (where $A \in N$ and $\sigma \in \Sigma$).
3) $S \to e$ (where S is the start state).

Example of Chomsky Normal Form

Recall the CFG:

$$S \rightarrow aaaaaaaa$$

Example of Chomsky Normal Form

Recall the CFG:

$$S \rightarrow aaaaaaaa$$

DISCUSS TO FIND A CHOMSKY NORMAL FORM CFG FOR $\{aaaaaaaa\}$.

Example of Chomsky Normal Form

Recall the CFG:
$S \rightarrow aaaaaaaa$

Example of Chomsky Normal Form

Recall the CFG:
$S \rightarrow aaaaaaaa$

Chomsky Normal form CFG that generates same lang:
$S \rightarrow AA$

Example of Chomsky Normal Form

Recall the CFG:

$S \to aaaaaaaa$

Chomsky Normal form CFG that generates same lang:

$S \to AA$

$A \to BB$

Example of Chomsky Normal Form

Recall the CFG:

$S \rightarrow aaaaaaaa$

Chomsky Normal form CFG that generates same lang:

$S \rightarrow AA$

$A \rightarrow BB$

$B \rightarrow CC$

Example of Chomsky Normal Form

Recall the CFG:

$S \rightarrow aaaaaaaa$

Chomsky Normal form CFG that generates same lang:

$S \rightarrow AA$

$A \rightarrow BB$

$B \rightarrow CC$

$C \rightarrow a$

Example of Chomsky Normal Form

Recall the CFG:

$S \rightarrow aaaaaaaa$

Chomsky Normal form CFG that generates same lang:

$S \rightarrow AA$

$A \rightarrow BB$

$B \rightarrow CC$

$C \rightarrow a$

We measure the size of a Chomsky Normal Form CFG by the number of rules.

Example of Chomsky Normal Form

Recall the CFG:

$S \rightarrow aaaaaaaa$

Chomsky Normal form CFG that generates same lang:

$S \rightarrow AA$

$A \rightarrow BB$

$B \rightarrow CC$

$C \rightarrow a$

We measure the size of a Chomsky Normal Form CFG by the number of rules.

So $\{aaaaaaaa\}$ has a Chomsky Normal Form CFG of size 4.

Chomsky Normal Form CFG for $\{a^n\}$

We say that $\{a^8\}$ has a CNF CFG of size 4.

Chomsky Normal Form CFG for $\{a^n\}$

We say that $\{a^8\}$ has a CNF CFG of size 4.

What about $\{a^{16}\}$? Vote

Chomsky Normal Form CFG for $\{a^n\}$

We say that $\{a^8\}$ has a CNF CFG of size 4.

What about $\{a^{16}\}$? Vote

1) Size 8

2) Size 5

Chomsky Normal Form CFG for $\{a^n\}$

We say that $\{a^8\}$ has a CNF CFG of size 4.

What about $\{a^{16}\}$? Vote
1) Size 8
2) Size 5
The answer is 5. Next slide.

Chomsky Normal Form CFG for $\{a^{16}\}$

$S \rightarrow AA$

Chomsky Normal Form CFG for $\{a^{16}\}$

$S \to AA$

$A \to BB$

Chomsky Normal Form CFG for $\{a^{16}\}$

$$S \to AA$$
$$A \to BB$$
$$B \to CC$$

Chomsky Normal Form CFG for $\{a^{16}\}$

$$S \rightarrow AA$$
$$A \rightarrow BB$$
$$B \rightarrow CC$$
$$C \rightarrow DD$$

Chomsky Normal Form CFG for $\{a^{16}\}$

$$S \to AA$$
$$A \to BB$$
$$B \to CC$$
$$C \to DD$$
$$D \to a$$

Chomsky Normal Form CFG for $\{a^{16}\}$

$$S \rightarrow AA$$
$$A \rightarrow BB$$
$$B \rightarrow CC$$
$$C \rightarrow DD$$
$$D \rightarrow a$$

What to do if n is not a power of 2. HW.

$L = \{a^n\}$

Upshot

For $L_n = \{a^n\}$:

- Any DFA or NFA that recognizes L_n has $n + \Omega(1)$ states.
- There is a CFG that generates L_n with $O(\log n)$ rules.

Our Old Friend $L = \{a, b\}^* a \{a, b\}^n$

1) We showed that L requires a 2^{n+1} size DFA.

Our Old Friend $L = \{a,b\}^* a\{a,b\}^n$

1) We showed that L requires a 2^{n+1} size DFA.

2) We have an NFA of size $n+2$. There is no NFA of size n since then there would be a DFA of size $2^n < 2^{n+1}$.

Our Old Friend $L = \{a, b\}^* a\{a, b\}^n$

1) We showed that L requires a 2^{n+1} size DFA.

2) We have an NFA of size $n+2$. There is no NFA of size n since then there would be a DFA of size $2^n < 2^{n+1}$.

3) DISCUSS for getting a CFG of size $\ll n$.

DFA, NFA, CFG

DFA, NFA, CFG

$L = L_1 \cdot L_2$ where

DFA, NFA, CFG

$L = L_1 \cdot L_2$ where

$L_1 = \{a, b\}^* a$. Has 5-rule Chomsky Normal Form CFG:
$S \rightarrow AS \mid BS \mid a$
$A \rightarrow a$
$B \rightarrow b$

DFA, NFA, CFG

$L = L_1 \cdot L_2$ where

$L_1 = \{a, b\}^* a$. Has 5-rule Chomsky Normal Form CFG:
$S \to AS \mid BS \mid a$
$A \to a$
$B \to b$

$L_2 = \{a, b\}^n$. A $\lg(n) + O(1)$ rule Chomsky Normal Form CFG.
$S \to S_1 S_1$
$S_1 \to S_2 S_2$
$\vdots$
$S_{\lg(n)-1} \to S_{\lg(n)} S_{\lg(n)}$
$S_{\lg(n)} \to a \mid b$
Note We are assuming n is a power of 2.

DFA, NFA, CFG Size Diff

$$L = \{a, b\}^* a \{a, b\}^n$$

DFA, NFA, CFG Size Diff

$$L = \{a, b\}^* a \{a, b\}^n$$

1) DFA of size $\Theta(2^n)$.

DFA, NFA, CFG Size Diff

$$L = \{a, b\}^* a\{a, b\}^n$$

1) DFA of size $\Theta(2^n)$.
2) NFA of size $n + \Theta(1)$.

DFA, NFA, CFG Size Diff

$$L = \{a, b\}^* a\{a, b\}^n$$

1) DFA of size $\Theta(2^n)$.
2) NFA of size $n + \Theta(1)$.
3) CFG of size $\Theta(\lg(n))$.

Any CFG can be Put Into Chomsky Normal Form

Recall the CFG for $\{a^m b^n : m > n\}$. We put it into Chomsky Normal Form.

1) $S \to AT$
2) $T \to aTb$
3) $T \to e$
4) $A \to Aa$
5) $A \to a$

Any CFG can be Put Into Chomsky Normal Form

Recall the CFG for $\{a^m b^n : m > n\}$. We put it into Chomsky Normal Form.
1) $S \to AT$
2) $T \to aTb$
3) $T \to e$
4) $A \to Aa$
5) $A \to a$
New nonterminals $[aT]$, $[b]$, $[a]$. Replace $T \to aTb$ with:
$T \to [aT][b]$
$[aT] \to [a]T$
$[b] \to b$.
$[a] \to a$

Any CFG can be Put Into Chomsky Normal Form

Recall the CFG for $\{a^m b^n : m > n\}$. We put it into Chomsky Normal Form.
1) $S \to AT$
2) $T \to aTb$
3) $T \to e$
4) $A \to Aa$
5) $A \to a$
New nonterminals $[aT]$, $[b]$, $[a]$. Replace $T \to aTb$ with:
$T \to [aT][b]$
$[aT] \to [a]T$
$[b] \to b$.
$[a] \to a$
Repeat the process with the other rules.

MISC

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.
3) Recall: DFA's are **Recognizers**, Regex are **Generators**.
CFG's are **Generators**. There is a **Recognizer** equivalent to it:

PDAs

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.
3) Recall: DFA's are **Recognizers**, Regex are **Generators**.
CFG's are **Generators**. There is a **Recognizer** equivalent to it:

PDAs

PDA does not stand for **Public Display of Affection**

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.
3) Recall: DFA's are **Recognizers**, Regex are **Generators**.
CFG's are **Generators**. There is a **Recognizer** equivalent to it:

PDAs

PDA does not stand for **Public Display of Affection**
PDA does stand for **Push Down Automata**.
They are NFAs with a stack.

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.
3) Recall: DFA's are **Recognizers**, Regex are **Generators**.
CFG's are **Generators**. There is a **Recognizer** equivalent to it:

PDAs

PDA does not stand for **Public Display of Affection**
PDA does stand for **Push Down Automata**.
They are NFAs with a stack.
Deterministic CFG's are **defined** by DPDA's where are DFAs with a stack.

MISC

1) There is a pumping theorem for CFL's but we won't be doing it.
2) If L_1 is a CFL and L_2 is regular then $L_1 \cap L_2$ is a CFL.
3) Recall: DFA's are **Recognizers**, Regex are **Generators**.
CFG's are **Generators**. There is a **Recognizer** equivalent to it:

PDAs

PDA does not stand for **Public Display of Affection**
PDA does stand for **Push Down Automata**.
They are NFAs with a stack.
Deterministic CFG's are **defined** by DPDA's where are DFAs with a stack.
The proof that PDA-recognizers and CFG-generators are equivalent is messy so we won't be doing it. We won't deal with PDA's in this course at all.