Announcements

Pset 0 graded
Pset 1 due tomorrow night
Pset 2 out
Midterm #1 Wednesday

Pset 1 subsequences
Q4

if x is a subseq of w
and $x = x_1 \cdots x_n$ $\quad w = w_1 \cdots w_m$

$$\cdots x_1 \cdots x_2 x_3 \underbrace{\cdots}_{\Sigma^*} x_4 \cdots$$

Midterm structure: 6 "long-answer" questions
(5 worth 30pts each, 1 15pt)
out of 150

Q1: 15 T/F
Q2-5: covers material seen in class
Q6: proof based on existing proofs from class

Format:

4 unstapled pieces of paper (both sides)

return: unstapled

Name & student id are preprinted on all sheets both sides

Regular Grammars

A regular grammar is a —tuple (V, Σ, R, S)

- V = finite set of variables (non terminals)
- Σ = finite set of terminals
- R = finite set of rules
- $S \in V$ - start variable

rules are of the form:

1. $A \to \varepsilon$ 2. $A \to a$ $(a \in \Sigma)$

left side is always a single variable

3. $A \to B$ 4. $A \to aB$

$B \in V$

Ex: $S \to 0S$ $\Sigma = \{0, 1\}$

(s is start) $S \to T$ $V = \{S, T\}$

$T \to 1T \mid \varepsilon$

represents $T \to 1T$
$T \to \varepsilon$

Derive the string
001

$S \Rightarrow 0S \Rightarrow 00S \Rightarrow 00T \Rightarrow 001T \Rightarrow 001$

language of grammar is $0^* 1^*$

yields (derives)

Say that $uAv \Rightarrow uxv$ (A variable)
if $A \to x$ is a rule

if $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n$
where $w_n \in \Sigma^*$ then we say
$w_n \in L(G)$

reg. Grammar G.

$S \Rightarrow^* w_n$ (0 or more rule applications to get w_n)

Claim: Every reg lang can be generated by a regular grammar.

If L is a reg lang, there is a DFA $D = (Q, \Sigma, \delta, q_0, F)$ for L.

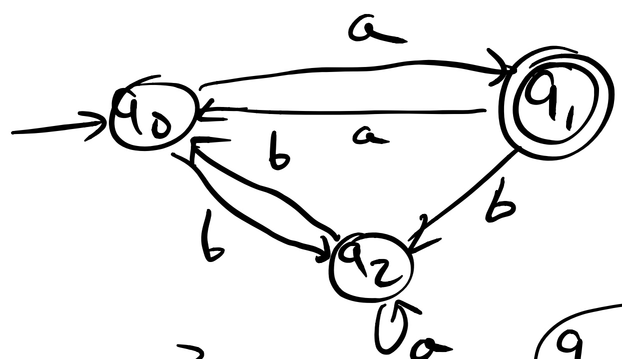
Make a variable q for every state $q \in Q$.

Start variable is q_0

$\delta(q, a) = q'$ make rule $q \rightarrow a q'$

for all $q \in F$ add rule $q \rightarrow \varepsilon$

Ex:



$V = \{q_0, q_1, q_2\}$

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

$S = q_0$

$q_0 \rightarrow a q_1$

$q_0 \rightarrow b q_2$

$q_1 \rightarrow a q_0 \mid b q_2 \mid \varepsilon$

$q_2 \rightarrow a q_2 \mid b q_0$

Claim: Every language generated by a reg grammar is regular.

1. if $A \to aB$ then $A \xrightarrow{a} B$
2. if $A \to B$ then $A \xrightarrow{\varepsilon} B$
3. if $A \to a$ then $A \xrightarrow{a}$ new final state
4. if $A \to \varepsilon$ then $A \xrightarrow{\varepsilon}$ new final state

is the only final state

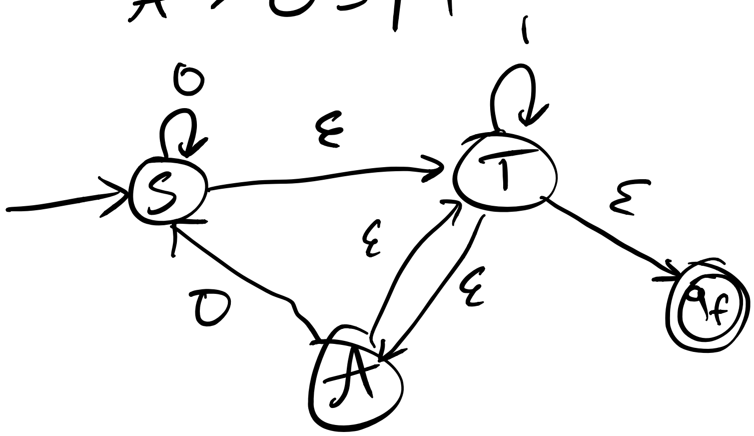
Ex: (start S)

$S \to 0S$

$S \to T$

$T \to 1T \mid \varepsilon \mid A$

$A \to 0S \mid T$



$S \to 0S1 \mid \varepsilon$ $\quad \Sigma_1 = \{0,1\}$

$V = \{S\}$

$L(G) = \{0^n 1^n : n \geq 0\}$.

Context-Free Grammars

Exactly the same as reg grammars except R has rules of the form

$A \to X$ where $A \in V$
$X \in (V \cup \Sigma_1)^*$

Ex: $A \to abcDEFFaABc$

$A \to \varepsilon$

Say that $w_1 \Rightarrow^* w_2$

if $w_1 = w_2$ or there exists w_3

s.t. $w_1 \Rightarrow w_3$ and $w_3 \Rightarrow^* w_2$.

The language of a context-free grammar G,

$L(G) = \{ w \in \Sigma_1^* : S \Rightarrow^* w \}$.

Then a language is a context-free language (CFL) if it is the language of some CFG.

Ex: $\{0^n 1^n : n \geq 0\}$ is a CFL.

Claim: Every regular language is context-free.

Proof: Every R.G. is a CFG.

The set of reg. langs form a proper subset of the set of CFLs.

Ex: $S \to aSa \mid bSb \mid \varepsilon \mid a \mid b$ $\quad \Sigma = \{a, b\}$

"palindromes"

$L(G) = \{ww^R : w \in \{a,b\}^*\}$.

Derive ababa

$L(G) = \{w \in \{a,b\}^* : w = w^R\}$.

Choose $s = a^p b^p b^p a^p$.

$$S \to (S) \mid SS \mid \varepsilon \qquad \Sigma = \{(,)\}$$

"balanced parentheses"

$$\underbrace{(())}_{S}\underbrace{()}_{S}$$

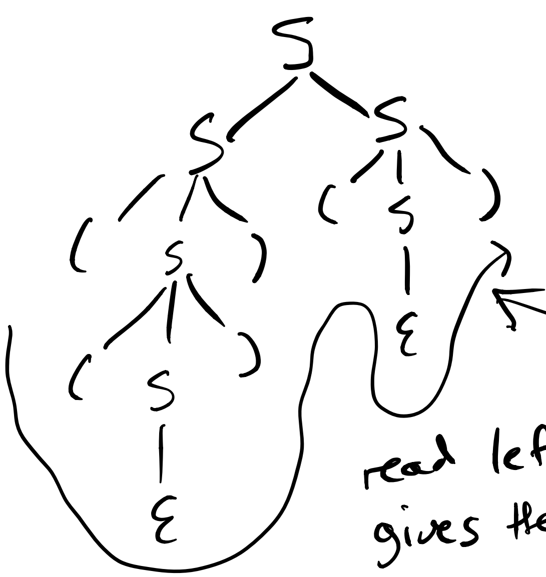
$$S \Rightarrow SS \Rightarrow S(S) \Rightarrow (S)(S) \Rightarrow (S)() \Rightarrow ((S))() \Rightarrow (())()$$

ε

$S \Rightarrow \varepsilon$

$S \Rightarrow SS \Rightarrow S \Rightarrow \varepsilon$

A <u>leftmost</u> derivation is a derivation where the leftmost variable is applied with a rule.



parse tree

read left-to-right gives the derived string

$$\underbrace{(())}_{S}\underbrace{()}_{S}$$

leftmost derivations & parse trees are in 1-1 correspondence.

Call a grammar **ambiguous** if there is **some** string in its language having ≥ 2 different leftmost derivations/parse trees.

Unambiguous CFG for balanced parentheses:

$$S \to \varepsilon \mid (S)S$$

Chomsky Normal Form

A CFG is in **Chomsky Normal Form** (CNF) if all rules are of the form

$$S \to \varepsilon$$
$$A \to a \quad (A \in V)$$
$$A \to BC \quad B, C \neq S$$

$$S \to (S) \mid SS \mid \varepsilon$$

Step 1: (of 5) Make a new start variable S_0 with rule $S_0 \to S$

$$S_0 \to S$$
$$S \to (S) \mid SS \mid \varepsilon$$

Step 2:

Office Hours/
Problem Solving

PS 1 Q#1:

Q2: 12022

Q4: Substring vs subsequence

subseq. is a sequence of indices $i_1 < i_2 < \dots < i_k$

Substring has $i_1 = i_2 - 1,\ i_2 = i_3 - 1, \dots$

a$\underline{abccd}$abc

bdbc a substring? X

a subsequence? ✓

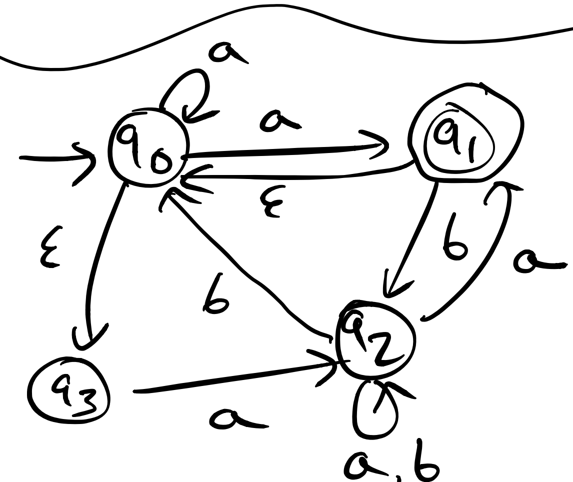
abccd a substring? ✓

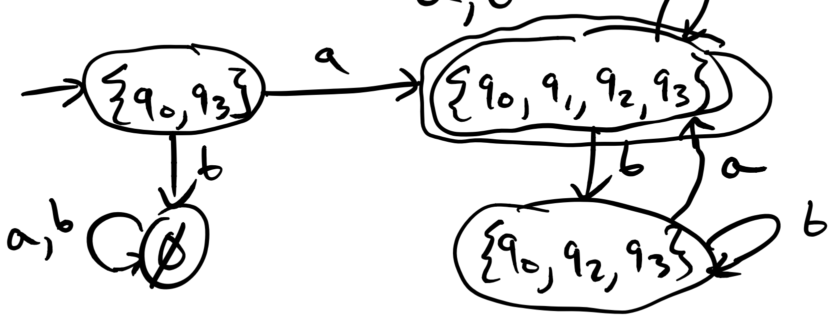
a subsequence? ✓

$w_1 w_2 w_3 \dots w_n \in L$

$\dots w_1 \dots w_2$

NFA:





$L = \{ w \in \{0,1,2\}^* :$ w has an equal 01 substrings than 10 substrings$\}$.

Claim: L is not regular.

Assume L is regular.

$\Rightarrow \exists$ a p for L.

Choose $w = (01)^p 2 (10)^p$ ~~X~~

0101$\underline{01}$0101

$w = (012)^p (210)^p$ $(w = xyz)$

$01\underline{\cancel{2}}012$ - - -

$\underline{0}1\underline{2}\underline{01}2$ - - - - -

No matter the decomposition of w into xyz s.t. $|y| \geq 1$ and $|xy| \leq p$

consider $xy^0 z = xz$

Removing any char (except for y=2) on the "first half" of the string removes at least 1 01 substring, but does not change the # of 10 substrings.

if y=2, then # of 10 substrings ↑, but # of 01 stays the same.

$\Rightarrow xz \notin L$, so L is not reg.

$L = \{w \text{ contains } 01 \text{ as a substring}\} \cup \{w : |w| \text{ is odd}\}$

$\downarrow$ reg (NFA)

$\downarrow$ reg (DFA)

$$L = \begin{cases} 1 & \text{if life is ever found on Mars} \\ 0 & \text{otherwise} \end{cases}$$