CS 154 Problem Session

April 19. 2002

1. **Review.** Suppose L_z, L_2, L_3, ... are each regular languages. Using what we know of regular operators and closure properties, can you conclude that $L_1 \quad L_2 \quad L_3 \quad$ ... is a regular language?

Answer: Consider $L_1 = \varepsilon$, $L_2 = 01$, $L_3 = 0011$, $L_4 = 000111$, and so on. Note each language L_i is regular (since it is finite); however it is easy to see that the union of all such languages is $L = \{ w \mid w \quad 0^n1^n, n \geq 0 \}$, which we have previously shown to be non-regular (via the Pumping Lemma for regular languages). In fact, since we may write a CFG to describe this language, we may conclude that L is a context free language.

2. **State Minimization**. Consider the DFA defined as follows:

	0	**1**
→A	B	E
B	C	F
***C**	D	H
D	E	H
E	F	I
***F**	G	B
G	H	B
H	I	C
***I**	A	E

Thus A is the initial state, and C, F, and I are the final states. Minimize this DFA.

Answer: We may find the equivalent minimized DFA using the table filling algorithm covered in HMU and in discussion section. We begin by noting that all final states and non-final states cannot be equivalent, and proceed to establish potential equivalences between the remaining states. Then, we iteratively eliminate all potential equivalences that depend on equivalences we have shown cannot hold. The result is the following table:

B	~~BC, EF~~							
C	✗	✗						
D	BE, EH	~~CE, FH~~	✗					
E	~~BF, EI~~	CF, FI	✗	~~EF, HI~~				
F	✗	✗	DG,BH	✗	✗			
G	BH, BE	~~CH, FB~~	✗	EH,BH	~~FH, BI~~	✗		
H	~~BI, CE~~	CI, CF	✗	~~EI, CH~~	FI, IC	✗	~~HI, BC~~	
I	✗	✗	AD,EH	✗	✗	AG,BE	✗	✗
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**

The remaining squares cannot be shown to be non-equivalent (they reference each other) and thus we will assume that they are equivalent. Hence we obtain that states AD, AG, BE, BH, CF, CI, DG, EH, and FI. Since we may apply transitivity for the equivalence operator we end up with the equivalence classes: {A, D, G}, {B, E, H}, and {C, F, I}. Each class may be transformed into a state, and the transitions for the minimized DFA determined using the transition table above (i.e., on 0, A->B, so {A, D, G} -> {B, E, H}). Note {A, D, G} is the new start state and {C, F, I} is a final state.

3. **CFG Conversion**. Consider the following grammar:

$$\mathbf{S \rightarrow 0S0 \mid 1S1 \mid} \varepsilon$$

a) What is this language?

Answer: This is the language L = { x | x is of the form ww^R, where $|w| \geq 0$ }. Note each x L is a palindrome, but since all strings in L are even, L does not include all possible palindromes.

b) Convert this grammar into a pushdown automata.

Answer: The formulaic construction outlined in the text is sufficient in this case. This requires only three states (q_{start}, q_{loop}, and q_{accept}) Σ_ε = { 0, 1, ε } and Γ_ε = { 0, 1, S, \$, ε }. The following table describes the behavior of the PDA by defining (next state, top stack) given (current state, current input, current top of stack):

	0	**1**	ε		
	0	**1**	**S**	**$**	ε
q_{start}					q_{loop}, S\$
q_{loop}	q_{loop},ε	q_{loop},ε	q_{loop},0S0 q_{loop},1S1 q_{loop},ε	q_{accept}, ε	
q_{accept}					

4. **Chomsky Normal Form**. Convert the following grammar to Chomsky Normal Form, eliminating all non-generating non-terminals as well as unreachable non-terminals.

V = {**S, X, Y, Z, W**}

Σ = { **(,), +, =, 0, 1** }

S → X=X

X → (X+X) | Y | X

Y → 0Y | 1Y | 0Z | 1Z | W

Z → ε

Answer:

1) Add a new start symbol (to prevent **S** from ever appearing on the right hand side):

$$\mathbf{S_0 \rightarrow S}$$

2) Eliminate ε-productions:

Y → 0Y | 1Y | 0Z | 1Z | 0 | 1 | W

3) Remove unit rules:

$$\mathbf{X \rightarrow (X+X) \mid 0Y \mid 1Y \mid 0Z \mid 1Z \mid 0 \mid 1 \mid W}$$
$$\mathbf{S_0 \rightarrow X{=}X}$$

4) Remove non-generating non-terminals (W , Z, and S) and remove unreachable non-terminals (none):

$$\mathbf{X \rightarrow (X+X) \mid 0Y \mid 1Y \mid 0 \mid 1}$$
$$\mathbf{Y \rightarrow 0Y \mid 1Y \mid 0 \mid 1}$$

5) Add additional non-terminal symbols to finish the conversion. Note this is a mechanical conversion and further simplification may be possible. This corrects the error made in section in which the **D** in the $\mathbf{X \rightarrow CD}$ rule was inadvertently ignored.

$\mathbf{S_0 \rightarrow AX}$	$\mathbf{A \rightarrow XB}$
$\mathbf{B \rightarrow =}$	$\mathbf{X \rightarrow CD}$
$\mathbf{C \rightarrow EF}$	$\mathbf{E \rightarrow (}$
$\mathbf{F \rightarrow XG}$	$\mathbf{G \rightarrow +}$
$\mathbf{D \rightarrow XH}$	$\mathbf{H \rightarrow)}$
$\mathbf{X \rightarrow UY}$	$\mathbf{X \rightarrow VY}$
$\mathbf{X \rightarrow 0 \mid 1}$	$\mathbf{Y \rightarrow UY}$
$\mathbf{Y \rightarrow VY}$	$\mathbf{Y \rightarrow 0 \mid 1}$
$\mathbf{U \rightarrow 0}$	$\mathbf{V \rightarrow 1}$

5. **Miscellaneous**.

a) Design a PDA to accept the language L = { w | w $\{a + b + c\}^*$ and # of a's ≥ # b's + # c's }.

Answer: The solution presented here is simplified from the one covered during discussion section. We will use the following alphabets:

$$\Sigma_\varepsilon = \{ \mathbf{a, b, c,} \ \varepsilon \}$$

Γ_ε = { **A, B, $, ** ε }

Q = { **X, Y, Z**}

Start state: **X**

Final state: **Z**

	a			**b OR c**			ε		
	A	**B**	**$**	**A**	**B**	**$**	**A**	**$**	ε
X									**Y, $**
Y	**Y, AA**	**Y,** ε	**Y, A$**	**Y,** ε	**Y, BB**	**Y, B$**	**Y,** ε	**Z,** ε	
Z									

b) How could you determine whether a CFG G generates an infinite number of strings?

Answer: This is a good question to think over as you prepare for the midterm. As a hint, think of converting the grammar to CNF form and eliminating all non-generating and unreachable non-terminals from the grammar. Thus the language of the new grammar G' is the same as the language of G. Now consider a tree with the start symbol S as its root and all possible expansions of S as the children of this node; for every non-terminal child of the root, repeat this procedure until a terminal is reached. Of interest will be an expansion chain from S that has more than n non-terminals present (where n is the number of non-terminals in G') prior to reaching a terminal symbol (at which point the expansion stops).