Midterm, Quarter 2

1.

(For the sake of clarity, q0:=S, q1:=A, q2:=B, q3:=C)

i	j	k	Operations/cases
1			
1	0	0	(i) True. **Add arc from 0 to 0 labeled *aa*.** (ii) False. (iii) False.
1	0	2	(i) False (ii) False (iii) False
1	0	3	(i) True. **Add arc from 0 to 3 labeled *ab*.** (ii) False. (iii) False.
1	2	0	(i) False. (ii) False. (iii) False.
1	2	2	(i) False. (ii) False. (iii) False.
1	2	3	(i) False. (ii) False. (iii) False.
1	3	0	(i) True. **Add an arc from 3 to 0 labeled *ba***.
1	3	2	(i) False. (ii) False. (iii) False.
1	3	3	(i) True. **Add an arc from 3 to 3 labeled *bb*.** (ii) False. (iii) False.
			Remove q1
3	0	0	(i) False. (ii) True. **Add an arc from 0 to 0 labeled *ab(bb)*ba.*** (iii) True. **Replace arcs from 0 to 0 with one labeled *ab(bb)*ba U aa.***
3	0	2	(i) False. (ii) True. **Add an arc from 0 to 2 labeled *ab(bb)*a.*** (iii) True. **Replace arcs from 0 to 2 with one labeled *ab(bb)*a U b.***
3	2	0	(i) False. (ii) True. **Add an arc from 2 to 0 labeled *a(bb)*ba*.** (iii) True. **Replace arcs from 2 to 0 with one labeled *a(bb)*ba U b***
3	2	2	(i) False. (ii) True. **Add an arc from 2 to 2 labeled *a(bb)*a*. (iii) False.**

Resulting grammar:

	ab(bb)*ba U aa	***ab(bb)*a U b***	***a(bb)*ba U b***	***a(bb)*a***
q0	{q0}	{q2}	{}	{}
q2	{}	{}	{q0}	{q2}

Final expression:

((ab(bb)*ba) U aa)*((ab(bb)*a) U b)((a(bb)*a) U((a(bb)*ba) U b)
((ab(bb)*ba)Uaa)*(ab(bb)*a) U b))*

2.

(a)

M: $Q = \{S,A,Z\}$, $\Sigma = \{a,b\}$, $F = \{Z\}$
$\delta(S,a)=S$, $\delta(S,b)=A$, $\delta(S,a)=Z$
$\delta(A,a)=S$, $\delta(A,b)=A$, $\delta(A,b)=Z$

(b)

Q' := {{S}}

X	**Current symbol**	**Y**	**New transition**
{S}	a	{S,Z}	δ({S}, a) = {S,Z}
{S}	b	{A}	δ({S}, b) = {A}
{S,Z}	a	{S}	δ({S,Z},a) = {S}
{S,Z}	b	{A}	δ({S,Z},b) = {A}
{A}	a	{S}	δ({A}, a) = {S}
{A}	b	{A,Z}	δ({A}, b) = {A,Z}
{A,Z}	a	{S}	δ({A,Z},a) = {S}
{A,Z}	b	{A,Z}	δ({A,Z},b) = {A,Z}

M': $Q = \{\{S\}, \{S,Z\}, \{A\}, \{A,Z\}\}$, $\Sigma = \{a,b\}$, $F = \{\{S,Z\},\{A,Z\}\}$
$\delta(\{S\}, a) = \{S,Z\}$
$\delta(\{S,Z\},a) = \{S\}$
$\delta(\{S,Z\},b) = \{A\}$
$\delta(\{A\}, a) = \{S\}$
$\delta(\{A\}, b) = \{A,Z\}$
$\delta(\{A,Z\},a) = \{S\}$

(c)

Theorem 6.3.2
$S \rightarrow aS \mid bA \mid aZ$
$A \rightarrow aS \mid bA \mid bZ$
$Z \rightarrow \lambda$

(d)

Theorem 6.3.2
$\{S\} \rightarrow a\{S,Z\}$
$\{S,Z\} \rightarrow a\{S\} \mid \lambda$
$\{S,Z\} \rightarrow b\{A\} \mid \lambda$
$\{A\} \rightarrow a\{S\}$
$\{A\} \rightarrow b\{A,Z\}$
$\{A,Z\} \rightarrow a\{S\} \mid \lambda$

(e)

(I did this one by hand to save time, using 6.2.2 on M) ((bb*a) U a)*((bb*b) U a)

3.

*Show that the language L = {a^{k^3} | k ε **N**} is not regular.*

Assume that L is regular. This implies that L is accepted by some DFA. Let k be the number of states of its DFA. By the pumping lemma, every string z ε L of length k or more can be decomposed into substring u, v, and w such that length(uv) <= k, v /= λ, and uviw ε L for all i >= 0.

Let us choose the arbitrary string a^{k^3} belonging to L with length greater than k. It can be written as uvw where the u, v, and w satisfy the conditions of the pumping lemma and 0 < length(v) <= k. If i = 3, we can proceed similarly to the proof in Example 6.6.1:

$$\begin{aligned} length(uv^3w) &= length(uvw) + length(v) + length(v) \\ &= k^3 + length(v) + length(v) \\ &<= k^3 + k + k < k^3 + 3k^2 + 3k + 1 = (k+1)^3 \end{aligned}$$

Therefore, the length of string uv^3w is in between the cubes of k and k+1, so it must not be a cube itself and is not in the language, which is a contradiction of our assumption.

4.

Show that left linear grammars generate precisely the regular sets.

Theorem 6.2.3 states that languages accepted by DFAs are regular languages, so we must show that L can be converted into a DFA. We can try to convert L into a right regular grammar and use 6.3.1, or we can convert it into a left regular grammar and modify 6.3.1 to show that left regular grammars can be converted into NFAs. I will try both.

First, it can be shown that a left linear grammar can be converted into a left regular grammar ($u_0u_1u_2$ ... u_n are the terminal characters of *u*):

A -> u can be converted to: A -> A_nu_n A -> $A_{n-1}u_{n-1}$ ... A -> u_0	A -> Bu can be converted to: A -> A_nu_n A -> $A_{n-1}u_{n-1}$ ... A -> Bu_0

Second, it can be shown that a left regular grammar can be converted into a right regular grammar ($w = b_0b_1b_2$ ... b_n *where B =>* w*)*:

A -> Ba can be converted to:
A -> b_0B_0
B_{10}-> b_1B_1
...
B_{n-1} -> b_nA'
A' -> a

Last, I will give a modification of theorem 6.3.1 for converting left regular grammars into NFAs. The conversions are basically a reversal of those in 6.3.1:

(i) Same as 6.3.1
(ii) d(B, a) = A when A -> Ba
d(Z, a) = A when A -> a

(iii) F = {S} where S is the start symbol of G
(iv) S = Z

5.

(a)

	λ λ/λ	b λ/B	a λ/A	b B/λ	a A/λ
q0	{q1}	{q0}	{q0}	{}	{}
q1	{}	{}	{}	{q1}	{q1}

(b)

[q0, ab, λ] |- [q0, b, A] |- [q1, b, A] |- [q1, λ, λ] success.	[q0, abb, λ] |- [q0, bb, A] |- [q1, bb, A] |- [q1, b, λ] fail.	[q0, abbb, λ] |- [q0, bbb, A] |- [q1, bbb, A] |- [q1, bb, λ] fail.

(c)

[q0, aaaa, λ] |- [q0, aaa, A] |- [q0, aa, AA] |- [q1, aa, AA] |- [q1, a, A] |- [q1, λ, λ] success.	[q0, baab, λ] |- [q0, aab, B] |- [q0, ab, AB] |- [q1, ab, AB] |- [q1, b, B] |- [q1, λ, λ] success.

(d)

[q0, aaa, λ] |- [q0, aa, A] |- [q0, a, AA] |- [q1, a, AA] |- [q1, λ, A] fail. (this pda could be easily modified to accept odd-length palindromes by adding a λ/λ and b λ/λ transitions from 0 to 1.)	[q0, ab, λ] |- [q0, b, A] |- [q1, b, A] |- [q1, b, A] fail.

6.

This PDA is based on the idea that the number of b's must be either less or greater than the number of a's. The arc to the state 'GT' is taken if j >i is desired, and the arc to the state 'LT' is taken if j<i is desired.

M : Q = {S, GT, LT}, Σ = {a,b}, Γ= {A}, F = {GT, LT}
δ(S, a, λ) = {[S, A]}
δ(S, b, λ) = {[GT, λ]}
δ(S, λ, A) = {[LT, λ]}
δ(GT, b, A) = {[GT, λ]}
δ(GT, b, λ) = {[GT, λ]}
δ(LT, b, A) = {[LT, λ]}
δ(LT, λ, A) = {[LT, λ]}

7.

(a)

Each 'a' pushes an A and forces another 'a', and each 'b' pops an A.

M: Q = {q0, q1, q2}, Σ = {a,b}, Γ = {A}, F = {q0, q2}

δ(q0, a, λ) = {[q1, A]}

δ(q0, b, A) = {[q2, λ]}

δ(q1, a, λ) = {[q0, λ]}

δ(q2, b, A) = {[q2, λ]}

(b)

See *(a)*

(c)

M: Q = {q0, q1}, Σ = {a,b}, Γ = {A}, F = {q1}

δ(q0, a, λ) = {[q0, A]}

δ(q0, b, AA) = {[q1, λ]}

δ(q1, b, AA) = {[q1, λ]}

(δ)

for M_1/M_2

[q0, aab, λ] |- [q1, ab, A] |- [q0, b, A] |- [q2, λ, λ] success.

for M_3

[q0, aab, λ] |- [q0, ab, A] |- [q0, b, AA] |- [q1, λ, λ] success.

8.

Assume L is context free. By Theorem 7.4.1, the string z = ww^Rw, where length(w) >=k and k is the number specified by the pumping lemma, can be decomposed into substrings *uvwxy* that satisfy the repetition properties. Consider the possible decompositions of z:

For the sake of clarity, let abc = ww^Rw where a = c = w and b = w^R. Let us consider the possibilities of which of the substrings a, b, and c can be lengthened.

(i) pumping s only lengthens b

(ii) pumping s lengthens a and b or b and c

(iii) pumping s lengthens a,b, and c

It is required that the length of a, b, and c be equal, and this would not be the case were only one or two substrings were lengthened in case (i) and (ii). In case (iii), we have lengthened each of a, b, and c equally. However, since the length(a) >= k, then length(vwx) <= length(a) = length(c), we cannot simultaneously lengthen the end of a and the end of c with the same pumps, which would be required for a and b to remain equal.W

9.

Initialize D and S to 0s and {}s:
Mark 1 in D at: [0,1], [0,3],[0,5],[0,6],[1,2],[1,4],[2,3],[2,5],[2,6],[3,4],[4,5],[4,6]

i	j	m	n	Operation
0	2	1	5	Add [0,2] to S[1,5]
0	2	3	5	Add [0,2] to S[3,5]
0	4	1	5	Add [0,4] to S[1,5]
0	4	3	5	Add [0,4] to S[3,5]
1	3	2	4	Add [1,3] to S[2,4]
1	5	2	6	DIST(1,5) D[1,5] := 1 DIST(0,2) D[0,2] := 1 DIST(0,4) D[0,4] := 1
1	6	2	6	DIST(1,6) D[1,6] := 1
2	4	5	5	
3	5	4	6	DIST(3,5) D[3,5] := 1 DIST(0,2) D[0,2] := 1 DIST(0,4) D[0,4] := 1
3	6	4	6	3.1 DIST(3,6) D[3,6] := 1

D:

	q0	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**
q0	0	1	1	1	1	1	1
q1	0	0	1	0	1	1	1
q2	0	0	0	1	0	1	1
q3	0	0	0	0	1	1	1
q4	0	0	0	0	0	1	1
q5	0	0	0	0	0	0	0
q6	0	0	0	0	0	0	0

S[1,5] = {[0,2],[0,4]}
S[2,4] = {[1,3]}
S[3,5] = {[0,2],[0,4]}

Otherwise empty

(b) Equivalence classes of M:

State	Equivalence Class
q0	λ
q1	a
q2	aa U bb
q3	b
q4	ab U ba
q5	(a U b)(a U b) (a U b)b*
q6	(a U b)(a U b) (a U b)b*a(a U b)*

(c)

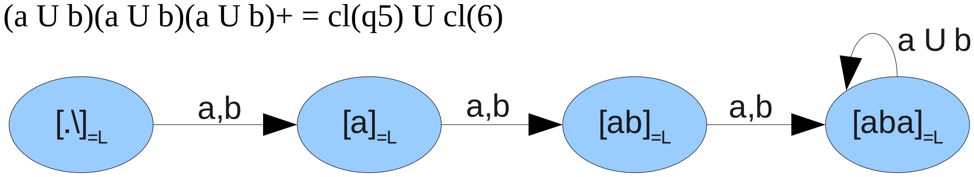
Equivalence classes of L

$[\lambda]_{=L}$ = λ = cl(q0)

$[a]_{=L}$ = a U b = cl(q1) U cl(q3)

$[aa]_{=L}$ = (a U b)(a U b) = cl(q2) U cl(q4)

$[aaa]_{=L}$ = (a U b)(a U b)(a U b)+ = cl(q5) U cl(6)



10.

(a) t1...tn denotes any ground term of L_C and c1...cn denotes a new constant

```
        F((Ax)(φ(x) & ψ(x)) <-> (Ax)φ(x) & (Ax)ψ(x))
          /                                       \
T((Ax)(φ(x) & ψ(x))                     F((Ax)(φ(x) & ψ(x))
          |                                  |
F((Ax)φ(x) & (Ax)ψ(x))                  T((Ax)φ(x) & (Ax)ψ(x))
          |                                  |
T((Ax)(φ(x) & ψ(x))                     F((Ax)(φ(x) & ψ(x))
          |                                  |
T(φ(t1) & ψ(t1))                        F(φ(c1) & ψ(c1))
          |                             /         \
     T(φ(t1))                      F(φ(c1))       F(ψ(c1))
          |                             |         |
     T(ψ(t1))          T((Ax)φ(x)&(Ax)ψ(x))      T((Ax)φ(x)&(Ax)ψ(x))
          |                             |         |
F((Ax)φ(x) & (Ax)ψ(x))           T((Ax)φ(x))     T((Ax)φ(x))
     /         \                        |         |
F(Ax)φ(x)      F(Ax)ψ(x)           T(φ(t1))      T(φ(t1))
     |         |                        |         |
F(φ(c1))       F(ψ(c1))                 X        T((Ax)ψ(x))
     |         |                                  |
     X         X                                 T(ψ(t2))
                                                  |
                                                  X
```

(b)

```
                    F((Ex)~φ(x) -> ~(Ax)φ(x))
                              |
                         T((Ex)~φ(x))
                              |
                         F(~(Ax)φ(x))
                              |
                         T((Ax)φ(x))
                              |
                         T(φ(t1))
                              |
                         T((Ex)~φ(x))
                              |
                         T(~φ(c1))
                              |
                         F(φ(c1))
                              |
                              X
```

11.

By assumption, we have a tree T with $<_L$, the left-right ordering on each level, which is well-ordered. By the definition of a tree (1.3), we also have the level ordering, $<_T$, which is well-ordered. Let every node of our tree N1, N2 ... Nn, be represented in a set of pairs of natural numbers, {(a,b)}, where a = lv(Nn) and b = lr(Nn). The function 'lr' is the ordinal value of the node's left-right ordering within its level, and the function 'lv' the ordinal value of its level. The definition of LL ordering is then: (a,b) ≤ (c,d) <=> a < c v (a = c & b≤ d).

Let us take the nodes Nx, Ny, and Nz, and show that the $<=_{LL}$ ordering is a partial order. If lv(Nx) < lv(Ny) and lv(Ny) < lv(Nz), then lv(Nx) < lv(Nz) by the well ordering of the natural numbers and Nx $<=_{LL}$ Nz.

If both lv(Nx) = lv(Ny) and lv(Ny) = lv(Nz), then the partial order is on lr: If lr(Nx) <= lr(Ny) and lr(Ny) <= lr(Nz) then lr(Nx) <= lr(Ny) by the ordering of the natural numbers, so Nx $<=_{LL}$ Nz.

If lv(Nx) = lv(Ny), then we can say the following: If lr(Nx) <= lr(Ny) & lv(Nx) = lv(Ny) & lv(Ny) < lv(Nz), then Nx $<=_{LL}$ Nz.

Similarly, if lv(Ny) = lv(Nz), then we can say: If lv(Nx) < lv(Ny) & lr(Ny) <= lr(Nz) then Nx $<=_{LL}$ Nz.

We must now show that the ordering also satisfies the trichotomy law. Let us take any two nodes, Nx and Ny, and show that their level combined with their left-right value obeys the trichotomy law by showing that our lexicographic operators are defined in terms of the ordering on natural numbers.

If Nx $<_{LL}$ Ny, then lv(Nx) < lv(Ny) or (lv(Nx) = lv(Ny) and lr(Nx) < lr(Ny)). The same can be said if x and y are switched, where y $<_{LL}$ x. If Nx $=_{LL}$ Ny then lv(Nx) = lv(Ny) and lr(Nx) = lr(Ny).

Now, we can show that the LL ordering of T is a well ordering. By definition 1.3, all trees have a least element at the 0th level, the root.

12.

(Referring to the cases in Figure 29 on p. 110)
Add “We choose to extend P_n accordingly” to the end of each of these.
2a: By induction, A_n |= a & b, so A_n |= a and A_n |= b.
2b: By induction, A_n |= ~(a & b), so A $_n$|= ~a or A_n |= ~b.
3a: By induction, A_n |= ~a, and no inductive step is required.
3b: By induction, A_n |= a, and no inductive step is required.
4b: By induction, A_n |= F(a v b), so A_n |= ~a and A_n |= ~b
5a: By induction, A_n |= a -> b, so A_n |= ~a or A_n |= b
5b: By induction, A_n |= ~(a -> b), so A_n |= a and A |= ~b
6a: By induction, A_n |= a <-> b, so either A_n |= a and A_n |= b or A_n |= ~a and A_n |= ~b
6b: By induction, A_n |= ~(a <-> b), so either A_n |= a and A_n|= ~b or A_n|=~a and A_n|=b
7a, 7b, 8a, 8b, and 4a covered by (i), (ii), and (iii) in lemma 7.1

13.

```
F(Ax(p(x,f(x))) -> AxEy(p(x,y)))
          |
T(Ax(p(x,f(x)))
          |
F(Ax(Ey(p(x,y))))
          |
F(Ey(p(c1,y)))
          |
F(p(c1,t1))
          |
T(Ax(p(x,f(x)))
          |
T(p(t2,f(t2)))      (f(t2)/t1 and c1/t2)
          |
          X

F(Ax(Ey(p(x,y))) -> Ax(p(x,f(x)))
          |
T(Ax(Ey(p(x,y))))
          |
F(Ax(p(x,f(x))))
          |
F(p(c1,f(c1)))
          |
T(Ax(Ey(p(x,y))))
          |
T(Ey(p(t1,y)))
          |
T(p(t1,c2))    f(c1) does not unify with c2 -- no conflict
```

14.

	~(AxEyP(x) -> ExEyR(x,y)) & Ax~EyQ(x,y)
3a'	Az(~(AxEyP(x) -> ExEyR(x,y)) & ~EyQ(z,y))
~(a -> b) = a & ~b	Az[(AxEyP(x) & ~(ExEyR(x,y))) & ~EyQ(z,y)]
Distribution of negations	Az[(AxEyP(x) & ~Ex~EyR(x,y)) & ~EyQ(z,y)]
Negation equivalencies from 9.1	Az[(AxEyP(x) & AxAy~R(x,y)) & Ay~Q(z,y)]
3a'	AzAw[(AxEyP(x) & AxAy~R(x,y)) & ~Q(z,w)]
3a'	AzAw[Au(AxEyP(x) & Ay~R(u,y)) & ~Q(z,w)]
3a'	AzAw[AuAv(AxEyP(x) & ~R(u,v)) & ~Q(z,w)]
3a	AzAw[AtAuAv(EyP(t) & ~R(u,v)) & ~Q(z,w)]
3b	AzAw[EsAtAuAv(P(s) & ~R(u,v)) & ~Q(z,w)]
3b	AwAzEsAtAuAv(P(s) & ~R(u,v)) & ~Q(z,w)
Skolemization:	AwAzAtAuAv(P(f(w,z)) & ~R(u,v)) & ~Q(z,w)

15.

D(Σ) = {x, g(v)}
Σ1 = {x/g(v)}
Σ1 = {Q(h(g(v),y),w), Q(h(g(v),a),f(v)), Q(h(g(v),a),f(b))}
D(Σ2) = {y, a}
Σ2 = {y/a}
Σ3 = {Q(h(g(v),a),w), Q(h(g(v),a),f(v)), Q(h(g(v),a),f(b))}
D(Σ3) = {w,f(v),f(b)}
Σ3 = {w/f(v)}
Σ4 = {Q(h(g(v),a),f(v)), Q(h(g(v),a),f(v)), Q(h(g(v),a),f(b))}
D(Σ4) = {v,b}
Σ4 = {b/v}
Σ5 = {Q(h(g(v),a),f(v)), Q(h(g(v),a),f(v)), Q(h(g(v),a),f(v))}
    = {Q(h(g(v),a),f(v))}
mgu for Σ = {x/g(v), y/a, w/f(v), b/v}

16.

(a)
D(Σ) = {y,f(w),f(a)}
Σ1 = {y/f(w)}
Σ1 = {P(h(f(w)),a,z), P(h(f(w)),a,w), P(h(f(a)),a,u)}
D(Σ1) = {w, a}
Σ2 = {a/w}
Σ2 = {P(h(f(w)),a,z), P(h(f(w)),a,w), P(h(f(w)),a,u)}
D(Σ2) = {z,w,u}

$\Sigma 3 = \{z/w\}$
$\Sigma 3 = \{P(h(f(w)),a,w), P(h(f(w)),a,w), P(h(f(w)),a,u)\}$
$D(\Sigma 3) = \{w,u\}$
$\Sigma 4 = \{u/w\}$
$\Sigma 4 = \{P(h(f(w)),a,w), P(h(f(w)),a,w), P(h(f(w)),a,w)\}$
$\quad = \{P(h(f(w)),a,w)\}$
mgu of $\Sigma = \{y/f(w), a/w, z/w, u/w\}$

(b)
$\Sigma = \{P(h(y),a,z), P(h(f(w)),a,w), P(h(f(a)),a,b)\}$
$D(\Sigma) = \{y,f(w),f(a)\}$
$\Sigma 1 = \{y/f(w)\}$
$\Sigma 1 = \{P(h(f(w)),a,z), P(h(f(w)),a,w), P(h(f(a)),a,b)\}$
$D(\Sigma 1) = \{w,a\}$
$\Sigma 2 = \{w/a\}$
$\Sigma 2 = \{P(h(f(w)),a,z), P(h(f(w)),a,w), P(h(f(w)),a,b)\}$
$D(\Sigma 2) = \{z,w,b\}$
$\Sigma 3 = \{z/w\}$
$\Sigma 3 = \{P(h(f(w)),a,w), P(h(f(w)),a,w), P(h(f(w)),a,b)\}$
$D(\Sigma 3) = \{w,b\}$
$\Sigma 4 = \{b/w\}$
$\Sigma 4 = \{P(h(f(w)),a,w), P(h(f(w)),a,w), P(h(f(w)),a,w)\}$
$\quad = \{P(h(f(w)),a,w)\}$
mgu of $\Sigma = \{y/f(w), w/a, z/w, b/w\}$