Quiz 1, Propositional Logic

Date: February 2

1. Prove $(((-p \rightarrow q) \land -q) \rightarrow p)$ is a tautology

   (a) (3pts) by truth table.

   \[
   \begin{array}{cccccccc}
   p & q & -p & -q & -p \rightarrow q & (-p \rightarrow q) \land -q & \rightarrow p \\
   T & T & F & F & T & F & T \\
   T & F & F & T & T & T & T \\
   F & T & T & F & T & F & T \\
   F & F & T & F & F & T & T \\
   \end{array}
   \]

   (b) (4pts) by algebra.

   
   \[
   \begin{align*}
   ((-p \rightarrow q) \land -q) \rightarrow p & \equiv (((\neg p \lor q) \land \neg q) \rightarrow p) \quad \text{conditional law} \\
   & \equiv ((\neg (p \lor q) \land -q) \lor p) \quad \text{law of negation} \\
   & \equiv (\neg (p \lor q) \land -q) \lor p \quad \text{conditional law} \\
   & \equiv (\neg (p \lor q) \lor \neg q) \lor p \quad \text{DeMorgan’s law} \\
   & \equiv (\neg (p \lor q) \lor q) \lor p \quad \text{law of negation} \\
   & \equiv (\neg p \lor (q \lor p)) \lor p \quad \text{associativity} \\
   & \equiv (\neg p \lor q) \lor (p \lor q) \quad \text{commutativity} \\
   & \equiv TRUE \quad \text{excluded middle}
   \end{align*}
   \]
2. (3pts) Using only $p \uparrow q$, where $p \uparrow q \equiv \neg(p \land q)$, express $p \rightarrow q$.

Note that $p \uparrow q \equiv \neg p \lor \neg q$ by DeMorgan’s law, and for any variable $q$, $(q \uparrow q) \equiv \neg(q \land q) \equiv \neg q$. Thus, we have:

\[
p \rightarrow q \equiv \neg p \lor q \\
\equiv \neg p \lor \neg \neg q \\
\equiv p \uparrow \neg q \\
\equiv p \uparrow (q \uparrow q)
\]

Another equally valid solution is $p \uparrow (p \uparrow q)$. 
Quiz 2, Rules of Inference  
*Date: February 9*

1. *(4pts)* Prove \((\neg p \rightarrow q) \land \neg q \rightarrow p\).

1. \[\((\neg p \rightarrow q) \land \neg q\]\hspace{1cm} Assumption  
2. \[\neg p \rightarrow q\]\hspace{1cm} \land\ elimination 1  
3. \[\neg q\]\hspace{1cm} \land\ elimination 1  
4. \[\neg\neg p\]\hspace{1cm} Modus tollens 2,3  
5. \[p\]\hspace{1cm} Double negation 4  
6. \[((\neg p \rightarrow q) \land \neg q) \rightarrow p\]\hspace{1cm} \rightarrow\ introduction 1,5

2. *(6pts)* Prove \((p \rightarrow q) \leftrightarrow (\neg q \rightarrow p)\).

1. \[p \rightarrow q\]\hspace{1cm} Assumption  
2. \[\neg q\]\hspace{1cm} Assumption  
3. \[\neg p\]\hspace{1cm} Modus tollens 1,2  
4. \[\neg q \rightarrow \neg p\]\hspace{1cm} \rightarrow\ introduction 2,3  
5. \[(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)\]\hspace{1cm} \rightarrow\ introduction 1,4  
6. \[\neg q \rightarrow \neg p\]\hspace{1cm} Assumption  
7. \[p\]\hspace{1cm} Assumption  
8. \[\neg q\]\hspace{1cm} Assumption  
9. \[\neg p\]\hspace{1cm} Modus ponens 6,8  
10. \[FALSE\]\hspace{1cm} Contradiction 7,9  
11. \[\neg \neg q\]\hspace{1cm} Reduction to absurdity 8,10  
12. \[q\]\hspace{1cm} Double negation 11  
13. \[p \rightarrow q\]\hspace{1cm} \rightarrow\ introduction 7,12  
14. \[\neg q \rightarrow \neg p\]\hspace{1cm} \rightarrow\ introduction 7,12  
15. \[(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)\]\hspace{1cm} \leftrightarrow\ introduction 5,14
Quiz 3, Predicate Logic
Date: February 16

1. (6pts) For $A[1...n]$, assert that every element (i.e. value) occurs exactly twice.

“Every element occurs at least twice” (“for every element, there is a different element with the same value”):

$$\forall i \in I_n : \exists j \in I_n : i \neq j \land A[i] = A[j]$$

“Every element occurs exactly twice” (“every element occurs at least twice, and there are no additional elements with the same value as such a pair”):

$$\forall i \in I_n : \exists j \in I_n : i \neq j \land A[i] = A[j] \land (\neg \exists k \in I_n : k \neq i \land k \neq j \land A[k] = A[i])$$

2. (4pts) Evaluate $\forall x \in N : \exists y \in I : (x - y)^2 - y < 0$.

False. We may be tempted to choose $y = x$, resulting in the expression $(0)^2 - y < 0$, or $y > 0$. However, this fails when $x = y = 0$, which is part of $N$. Indeed, when $x = 0$, we are left with $(y)^2 - y < 0$, or $y^2 < y$, which is impossible to satisfy with integer $y$ values.
1. (6pts) Prove
\[ \sum_{i=0}^{n} x^i = \frac{x^{n+1} - 1}{x - 1}, \quad n \geq 0, \quad x \neq 1. \]

When \( n = 0 \), we have
\[ \sum_{i=0}^{0} x^i = x^0 = 1 = \frac{x^{0+1} - 1}{x - 1} = \frac{x - 1}{x - 1} = 1. \]

Assume that for some \( k \geq 0, \)
\[ \sum_{i=0}^{k} x^i = \frac{x^{k+1} - 1}{x - 1}, \quad x \neq 1 \]

We would like to prove the \( k + 1 \) case,
\[ \sum_{i=0}^{k+1} x^i = \frac{x^{(k+1)+1} - 1}{x - 1} \]

To do this, we begin with the left hand side, and substitute the inductive hypothesis,
\[ \sum_{i=0}^{k+1} x^i = x^{k+1} + \sum_{i=0}^{k} x^i = x^{k+1} + \frac{x^{k+1} - 1}{x - 1} \]
\[ = \frac{(x - 1)x^{k+1}}{x - 1} + \frac{x^{k+1} - 1}{x - 1} = \frac{x^{(k+1)+1} - x^{k+1} + x^{k+1} - 1}{x - 1} \]
\[ = \frac{x^{(k+1)+1} - 1}{x - 1} \]

This proves the inductive conclusion, thus by mathematical induction, the theorem is proved.
2. (4pts) Give a formal outline of the proof of $\forall i \in I^+_k : p(i)$.

1. $p(k)$  
   Base case

2. $[i \in I^+_k]$  
   Assumption

3. $[p(i)]$  
   Assumption

4. $p(i + 1)$  
   proof of IC

5. $p(i) \rightarrow p(i + 1)$  
   $\rightarrow$ introduction 3,4

6. $\forall i \in I^+_k : p(i) \rightarrow p(i + 1)$  
   $\forall$ introduction 2,5

7. $\forall i \in I^+_k : p(i)$  
   Mathematical induction 1,6
Quiz 5, Program Verification
Date: March 2

1. (3pts) State, (5pts) prove, and (2pts) apply the loop invariant.

\[
x \leftarrow 4 \\
i \leftarrow 1 \\
\textbf{while } i < n \textbf{ do} \\
x \leftarrow 6 \times x \\
i \leftarrow i + 1 \\
x \leftarrow x / 3
\]

Loop invariant: \( x = 2^{i+1} \land i \leq n \) (assuming \( n \) is at least 1)
After initialization, \( x = 4 \) and \( i = 1 \), so \( 4 = 2^{1+1} \land 1 \leq n \).

Proof and application (inference rule version; the application is the \textit{while} step at the very end):

\[
x = 2^{i+1} \land i \leq n \land i < n \quad \{x \leftarrow 6 \times x\} \quad x = 6 \times 2^{i+1} \land i \leq n \land i < n \\
x = 6 \times 2^{i+1} \land i \leq n \land i < n \quad \{i \leftarrow i + 1\} \quad x = 6 \times 2^i \land i \leq n \\
\hline
x = 6 \times 2^i \land i \leq n \quad \{x \leftarrow x / 3\} \quad x = 2 \times 2^i = 2^{i+1} \land i \leq n \\
\hline
x = 2^{i+1} \land i \leq n \land i < n \quad \{x \leftarrow 6 \times x; i \leftarrow i + 1; x \leftarrow x / 3\} \quad x = 2^{i+1} \land i \leq n \\
x = 2^{i+1} \land i \leq n \quad \{\text{while } i < n \textbf{ do } S\} \quad x = 2^{n+1} \land i \leq n \land \neg(i < n)
\]

Here, \( S \) represents the code block \( x \leftarrow 6 \times x; i \leftarrow i + 1; x \leftarrow x / 3 \).
Note that \( \neg(i < n) \equiv i \geq n \), and \( i \leq n \land i \geq n \) implies that \( i = n \), so at the end of the loop, \( x = 2^{n+1} \).
Proof and application (inline annotation version; the application is the final comment after the loop):

\begin{verbatim}
x := 4
i := 1
// x = 2^{i+1} \land i \leq n
while i < n do
  // x = 2^{i+1} \land i \leq n \land i < n
  x := 6 \times x
  // x = 6 \times 2^{i+1} \land i \leq n \land i < n
  i := i + 1
  // x = 6 \times 2^i \land i \leq n
  x := x/3
  // x = 2 \times 2^i = 2^{i+1} \land i \leq n
  // x = 2^{i+1} \land i \leq n \land \neg (i < n)
\end{verbatim}
Quiz 6, Mathematical Induction, Revisited

Date: March 9

1. (5pts) Informally prove by induction: \( n! > 3^{n-1} \) for all \( n \geq 5 \).

Base case, \( n = 5 \):
\[ n! = 5! = 120 > 3^4 = 81 \]

Inductive hypothesis, \( k \) case:
Assume that \( k! > 3^{k-1} \) for some \( k \geq 5 \)

Inductive conclusion, \( k + 1 \) case:
The goal is to prove that \( (k + 1)! > 3^{(k+1)-1} \)

Proof:
\( (k + 1)! = (k + 1)k! \)
By the inductive hypothesis, \( k! > 3^{k-1} \), so
\( (k + 1)k! > (k + 1)3^{k-1} \)

Since \( k \geq 5 \), it follows that \( k + 1 > 3 \), so
\( (k + 1)! > (k + 1)3^{k-1} > 3 \cdot 3^{k-1} = 3^{k-1+1} = 3^{(k+1)-1} \)
\( (k + 1)! > 3^{(k+1)-1} \)

Therefore, by mathematical induction, the statement is proved.
2. (5pts) $S_n = 2S_{n-1} + 1$ for all $n \geq 1$, and $S_0 = 0$.
   Prove informally $S_n = 2^n - 1$ for all $n \geq 0$.

   Base case, $n = 0$:
   $S_n = S_0 = 0 = 2^0 - 1 = 1 - 1$

   Inductive hypothesis, $k - 1$ case:
   Assume that $S_{k-1} = 2^{k-1} - 1$ for some $k \geq 1$

   Inductive conclusion, $k$ case:
   We want to prove that $S_k = 2^k - 1$

   Proof:
   Beginning with the left hand side, and applying the given relationship
   $S_n = 2S_{n-1} + 1$, we get $S_k = 2S_{k-1} + 1$

   Substituting the inductive hypothesis gives
   
   $S_k = 2(2^{k-1} - 1) + 1 = 2^{k-1+1} - 2 + 1 = 2^k - 1$
   $S_k = 2^k - 1$

   Thus, by mathematical induction, the theorem is proved.
Quiz 7, Regular Expressions
Date: March 30

1. (4pts) Give the strings generated of length 5, \((ab + a)a^*(b + ab)\).

\{abaab, aaaaab\}. At the beginning of the string, there is a choice of \(ab\) or \(a\), and at the end of the string, there is a choice of \(b\) or \(ab\). Everything in between this prefix and suffix must be an \(a\), and the number of \(a\)s is fixed by the restriction that the overall length must be 5. Thus, there are 4 possible strings, two of which become redundant once we notice that, for instance, \((a)aaa(b)\) is the same as \((a)aa(ab)\).

2. (6pts) Give a regular expression for \(\Sigma = \{a, b\}, L = \{x \mid x \text{ does not contain } abb\}\).

If a string in \(L\) contains any \(b\)s, then either the \(b\)s must appear isolated from other \(b\)s, or they must appear at the beginning of the string, otherwise we would have an \(abb\). If we begin with \((a + b)^*\), the set of all strings, we can modify it by forcing all \(b\)s to be preceded by an \(a\), \((a + ab)^*\), which will prevent two \(b\)s from appearing in a row. If we also allow for the possibility that the string begins with \(b\)s, we would get:

\(b^*(a + ab)^*\)
Quiz 8, Regular Grammars

Date: April 6

1. (6pts) Convert into a regular grammar with unit productions: $a^*b$.

Using $P_4$ as the final answer, the start symbol is $S_4$.

Note that use of the algorithm is required for this question, otherwise a very simple answer would be possible: $\{S \to aS, S \to bB, B \to \Lambda\}$. 
2. (4pts) Convert into a regular grammar without unit productions:
\{S \rightarrow aA, S \rightarrow B, B \rightarrow bB, B \rightarrow A, A \rightarrow \Lambda, B \rightarrow aS\}.

Solution:

\[
\begin{align*}
S &\rightarrow aA \\
S &\rightarrow B \\
\quad S &\rightarrow B \\
\quad S &\rightarrow A \\
S &\rightarrow B \\
S &\rightarrow aS \\
S &\rightarrow \Lambda
\end{align*}
\]

\[
\begin{align*}
B &\rightarrow bB \\
B &\rightarrow A \\
B &\rightarrow aS
\end{align*}
\]

\[
\begin{align*}
A &\rightarrow \Lambda
\end{align*}
\]
Quiz 9, Deterministic Regular Grammars
Date: April 13

1. (10pts) Convert into a deterministic regular grammar:
   \[ S \to aA, S \to aB, A \to aB, A \to bS, A \to \Lambda, B \to bA, B \to bB. \]

   Solution:

   \[
   \begin{array}{c|cc}
   & a & b \\ \hline
   S & A,B & - \\
   A & B & S \\
   B & - & A,B \\
   S,A & A,B & S \\
   S,B & A,B & A,B \\
   A,B & B & S,A,B \\
   S,A,B & A,B & S,A,B \\
   \end{array}
   \]

   \[
   \begin{align*}
   V_{\{S\}} & \to aV_{\{A,B\}} \\
   V_{\{S\}} & \to bV_{\emptyset} \\
   V_{\{A,B\}} & \to aV_{\{B\}} \\
   V_{\{A,B\}} & \to bV_{\{S,A,B\}} \\
   V_{\{B\}} & \to aV_{\emptyset} \\
   V_{\{B\}} & \to bV_{\{A,B\}} \\
   V_{\{S,A,B\}} & \to aV_{\{A,B\}} \\
   V_{\{S,A,B\}} & \to bV_{\{S,A,B\}} \\
   V_{\{A,B\}} & \to \Lambda \\
   V_{\{S,A,B\}} & \to \Lambda \\
   \end{align*}
   \]

   The problem asked for a deterministic regular grammar (such as the one above) and not a regular expression, but to avoid confusion, regular expression solutions were also accepted on this problem (this will not be the case on the exam). On the following pages are some regular expression solutions for this problem.
To remove \( A \) then \( B \) then \( S \), first add \( S' \), \( H \), and missing loopbacks.

\[
\begin{align*}
S' & \to S \\
S & \to aA & A & \to aB & B & \to bA & H & \to \Lambda \\
S & \to aB & A & \to bS & B & \to bB \\
S & \to S & A & \to H \\
& & A & \to A
\end{align*}
\]

Remove \( A \).

\[
\begin{align*}
S & \to aA / A \to A / A \to aB & : & S \to aaB \\
S & \to aA / A \to A / A \to bS & : & S \to abS \\
S & \to aA / A \to A / A \to H & : & S \to aH \\
B & \to bA / A \to A / A \to aB & : & B \to baB \\
B & \to bA / A \to A / A \to bS & : & B \to bbS \\
B & \to bA / A \to A / A \to H & : & B \to bH
\end{align*}
\]

After removing \( A \), the remaining productions are:

\[
\begin{align*}
S' & \to S \\
S & \to (a + aa)B \\
S & \to abS \\
S & \to aH \\
B & \to (b + ba)B \\
B & \to bbS \\
B & \to bH
\end{align*}
\]

Remove \( B \).

\[
\begin{align*}
S & \to (a + aa)B / B \to (b + ba)B / B \to bbS : S \to (a + aa)(b + ba)^*bbS \\
S & \to (a + aa)B / B \to (b + ba)B / B \to bH : S \to (a + aa)(b + ba)^*bH
\end{align*}
\]

After removing \( B \), the remaining productions are:

\[
\begin{align*}
S' & \to S \\
S & \to (ab + (a + aa)(b + ba)^*bb)S \\
H & \to \Lambda \\
S & \to (a + (a + aa)(b + ba)^*b)H
\end{align*}
\]

Remove \( S \).

\[
\begin{align*}
S' & \to S / S \to (ab+(a+aa)(b+ba)^*bb)S / S \to (a+(a+aa)(b+ba)^*b)H \\
\text{Regular expression: } & (ab + (a + aa)(b + ba)^*bb)^*(a + (a + aa)(b + ba)^*b)
\end{align*}
\]
To remove \( B \) then \( S \) then \( A \), first add \( S' \), \( H \), and missing loopbacks.

\[
\begin{align*}
S' & \rightarrow S & S & \rightarrow aA & A & \rightarrow aB & B & \rightarrow bA & H & \rightarrow \Lambda \\
S & \rightarrow aB & A & \rightarrow bS & B & \rightarrow bB \\
S & \rightarrow S & A & \rightarrow H \\
& & A & \rightarrow A \\
\end{align*}
\]

Remove \( B \).

\[
\begin{align*}
S & \rightarrow aB / B \rightarrow bB / B \rightarrow bA : S & \rightarrow ab^*bA \\
A & \rightarrow aB / B \rightarrow bB / B \rightarrow bA : A & \rightarrow ab^*bA \\
\end{align*}
\]

After removing \( B \), the remaining productions are:

\[
\begin{align*}
S' & \rightarrow S & S & \rightarrow (a+ab^*b)A & A & \rightarrow bS & H & \rightarrow \Lambda \\
S & \rightarrow S & A & \rightarrow H \\
& & A & \rightarrow ab^*bA \\
\end{align*}
\]

Remove \( S \).

\[
\begin{align*}
S' & \rightarrow S / S \rightarrow S / S \rightarrow (a + ab^*b)A : S' & \rightarrow (a + ab^*b)A \\
A & \rightarrow bS / S \rightarrow S / S \rightarrow (a + ab^*b)A : A & \rightarrow b(a + ab^*b)A \\
\end{align*}
\]

After removing \( S \), the remaining productions are:

\[
\begin{align*}
S' & \rightarrow (a + ab^*b)A & A & \rightarrow H \\
H & \rightarrow \Lambda & A & \rightarrow (ab^*b + b(a + ab^*b))A \\
\end{align*}
\]

Remove \( A \).

\[
\begin{align*}
S' & \rightarrow (a + ab^*b)A / A \rightarrow (ab^*b + b(a + ab^*b))A / A \rightarrow H \\
\end{align*}
\]

Regular expression: \((a + ab^*b)(ab^*b + b(a + ab^*b))^*\)
Quiz 10, Finite Automata
Date: April 20

1. (3pts) Give a DFA for $L = \{x \mid x \text{ contains } aabaa\}$.

2. (3pts) Give a DFA for $L = \{x \mid x \text{ does not contain } abb\}$.

3. (3pts) Give a NFA for $L = \{x \mid \text{ every } bb \text{ in } x \text{ is preceded by an } a\}$.
Example: $abb \in L$, $ab \in L$, $babb \in L$.

The problem becomes easy if we note that the only strings which are not in $L$ are strings which begin in $bb$. 

\[ a, b \]
\[ a \]
\[ b \]
Quiz 11, Properties of Finite Automata and Regular Languages

Date: April 27

1. (6pts) Convert the following RG to a FA:
   \[ S \rightarrow aA, \quad S \rightarrow bC, \quad A \rightarrow aA, \quad A \rightarrow bS, \quad B \rightarrow aA, \quad B \rightarrow bS, \quad B \rightarrow \Lambda, \]
   \[ C \rightarrow aB, \quad C \rightarrow bS, \quad C \rightarrow \Lambda. \]

   Solution:
   
   ![Diagram of FA]

2. (4pts) For a language \( L = \{ x \mid x = a^k b c d^k e, k > 0 \} \), give a set \( S \) and suffix \( z \) which would prove that it is not regular.

   Reasonable answers include:
   
   \[ S = \{ a^k \mid k > 0 \}, \quad z = b c d^k e \]
   \[ S = \{ a^k b \mid k > 0 \}, \quad z = c d^k e \]
   \[ S = \{ a^k b c \mid k > 0 \}, \quad z = d^k e \]

   A full proof was not asked for, but it could proceed as follows: If you pick any two different elements from \( S \) (let’s say \( x = a^i \) and \( y = a^j \) for the first solution line, \( i \neq j \)), then any machine which accepts language \( L \) would be able to tell that \( xz = a^i b c d^i e \in L \), while \( yz = a^j b c d^j e \notin L \). Since \( S \) is an infinite set, and each string in \( S \) is distinguishable by your machine, the machine must have an infinite number of states, and thus cannot be a FA. Therefore, \( L \) is not regular.
Quiz 12, Context Free Languages

Date: May 4

1. (7pts) Give a CFG for \( L = \{a^i b^j c^k \mid j < i + k \} \).

In order to preserve the inequality \( j < i + k \), for every \( b \) that is added, there must be at least one \( a \) or \( c \).

\[
S \rightarrow aT \mid Tc \\
T \rightarrow aT \mid Tc \mid AC \\
A \rightarrow aAb \mid \Lambda \\
C \rightarrow bCc \mid \Lambda 
\]

2. (3pts) What is the language generated by:

\( S \rightarrow SS \mid a \)

The shortest string that can be created is the string \( a \), by immediately converting \( S \rightarrow a \). Every action on a string will either increase the string’s length by 2 (\( S \rightarrow SS \)) or keep it the same size (\( S \rightarrow a \)). There is only one nonterminal, \( S \), and only one terminal that it can convert into, \( a \), therefore all symbols in a string must eventually become \( a \)’s.

Thus, the language is the set of all strings of \( a \)’s of odd length:

\( L = \{a^{2i+1} \mid i \in N\} \)