Staff Solutions to Midterm Exam September 24

Problem 1 (Irrational logarithm) (15 points).
Prove that \(\log_{12} 18\) is irrational.

Solution. Proof. Suppose to the contrary that
\[
\log_{12} 18 = \frac{m}{n}
\]
for some integers \(m, n\) where \(n > 0\). So we have
\[
12^{\log_{12} 18} = 12^{m/n} \quad \text{(raising 12 to equal powers),}
\]
\[
18 = 12^{m/n} \quad \text{(def of log\(_{12}\)),}
\]
\[
18^n = 12^m \quad \text{(raising both sides to the \(n\)th power).}
\]
\[
(2 \cdot 3^2)^n = (2^2 \cdot 3)^m \quad \text{(factoring 18 \& 12 into primes).}
\]
\[
2^n \cdot 3^{2n} = 2^{2m} \cdot 3^m. \quad (1)
\]
By the uniqueness of prime factorization, the powers of primes on the two sides of equation (1) must be equal. That is,
\[
n = 2m \quad \text{and} \quad 2n = m.
\]
This is only possible if \(m = n = 0\), a contradiction.

Problem 2 (Digital Circuits, Well Ordering) (15 points).
An \(n\)-bit AND-circuit has 0-1 valued inputs \(a_0, a_1, \ldots, a_{n-1}\) and one output \(c\) whose value will be
\[
c = a_0 \ \text{AND} \ a_1 \ \text{AND} \ \cdots \ \text{AND} \ a_{n-1}.
\]

There are various ways to design an \(n\)-bit AND-circuit. A serial design is simply a series of AND-gates, each with one input being a circuit input \(a_i\) and the other input being the output of the previous gate as shown in Figure 1.

We can also use a tree design. A 1-bit tree design is just a wire, that is \(c := a_1\). Assuming for simplicity that \(n\) is a power of two, an \(n\)-input tree circuit for \(n > 1\) simply consists of two \(n/2\)-input tree circuits whose outputs are AND’d to produce output \(c\), as in Figure 2. For example, a 4-bit tree design circuit is shown in Figure 3.

(a) How many AND-gates are in the \(n\)-input serial circuit?

Solution. \(n – 1\).

(b) Briefly explain why the tree circuit is exponentially faster than the serial circuit.
Figure 1  A serial AND-circuit.
Figure 2  An $n$-bit AND-tree circuit.

Figure 3  A 4-bit AND-tree circuit.
Solution. The “speed” or latency of a circuit is the largest number of gates on any path from an input to an output. For the $n$-input serial circuit, the longest such path is from $a_1$ to $c$, and it has $n - 1$ AND-gates. For the $n$-input tree circuit, the length of every path from an input to output $c$ is $\log_2 n - 1$, which is exponentially smaller than $n$.

(c) Assume $n$ is a power of two. Prove using the Well Ordering Principle that the $n$-input tree circuit has $n - 1$ AND-gates.

Solution. Suppose some $n$-bit tree circuit had a different number of AND-gates. By WOP, there is a least $m$ such that $m$ is a power of two and the $m$-bit tree circuit does not have $m - 1$ gates.

When $n - 1$, then $n - 1$ is zero. Since the 1-bit tree circuit has no gates, $n - 1$ is the correct formula when $n = 1$, so $m$ must be $> 1$. Also, $m$ is a power of two, so it must be divisible by 2. So $m/2$ is a smaller power of 2, and since $m$ is minimum number for which the $n - 1$ count fails, an $m/2$-input tree circuit must have $m/2 - 1$ AND-gates. But since the $m$-input tree is made out of two $m/2$-input trees plus one AND-gate, it has $(m/2 - 1) + (m/2 - 1) + 1 = m - 1$ AND-gates, contradicting the assumption that the $m$-input tree has a different number of gates. This contradiction implies that the $n$-bit tree circuit has $n - 1$ gates for every $n$ that is a power of 2.

Problem 3 (Satisfiability) (6 points).
Explain why a logical formula $P$ is satisfiable iff its negation $\neg(P)$ is not valid.

Solution. Essentially the same as the solution to 3.11

To prove the iff, we prove that the left hand statement implies the right hand one and vice-versa.

(left-to-right case): If $P$ is satisfiable, then $\neg(P)$ is not valid.

Proof. $P$ is true in an environment iff $\neg(P)$ is false in that environment. Since $P$ is satisfiable, it is true in some environment, which means that $\neg(P)$ is false in some environment. So not all environments make $\neg(P)$ true, which means that $\neg(P)$ is not valid.

(right-to-left case): If $\neg(P)$ is not valid, then $P$ is satisfiable.

Proof. If $\neg(P)$ is not valid, some environment makes it false, and therefore this environment make $P$ true. This means that $P$ that $P$ is satisfiable by definition.

Problem 4 (Truth tables, Cases) (12 points).

Claim. There are exactly two truth environments (assignments) for the variables $M, N, P, Q, R, S$ that satisfy the following formula:

\[
\begin{align*}
&\left(\overline{P} \lor Q\right) \text{AND} \left(\overline{Q} \lor R\right) \text{AND} \left(\overline{R} \lor S\right) \text{AND} \left(\overline{S} \lor P\right) \text{AND} M \text{AND} \overline{N}
\end{align*}
\]

clause (1) clause (2) clause (3) clause (4)

(a) This claim could be proved by truth-table. How many rows would the truth table have?

Solution. A truth table for a formula with 6 variables has $2^6 = 64$ rows.

(b) Instead of a truth-table, prove this claim with an argument by cases according to the truth value of $P$. 

Solution. Obviously $M$ must be true and $N$ must be false. Now we have:

Case 1 ($P$ is false): In order to have any chance of satisfying clause (4), $S$ must be false. Similarly, if $S$ is false, then in order to satisfy clause (3), $R$ must be false; similarly, $Q$ must be false.

Case 2 ($P$ is true): $Q$ must be true to make clause (1) true and have any chances of making the overall expression true. Similarly, if $Q$ is true, then $R$ must be true and if $R$ is true then $S$ is true.

Those arguments prove there are at most two satisfying truth environments, but we need to show the two environments we were left with actually satisfy the formula. This can be easily done, by plugging the values into the formula:

If all variables $P$, $Q$, $R$, $S$ are set to true, then since clause (1) has $Q$ clause (2) has $R$, clause (3) has $S$, and clause (4) has $P$, then every clause is satisfied, and the full AND-combination is satisfied. If all are false, then since clause (1) has $P$, clause (2) has $Q$, clause (3) has $R$ and clause (4) has $S$, then again every clause is satisfied and the overall proposition is satisfied. So both of those satisfy the proposition.

Problem 5 (Set Formulas) (12 points).
Predicate Formulas whose only predicate symbol is equality are called “pure equality” formulas. For example,

$$\forall x \forall y. x = y$$

is a pure equality formula. Its meaning is that there is exactly one element in the domain of discourse.\(^1\)

Another such formula is

$$\exists a \exists b \forall x. x = a \text{ OR } x = b.$$  \hfill (≤ 2-elements)

Its meaning is that there are at most two elements in the domain of discourse.

A formula that is not a pure equality formula is

$$x \leq y.$$  \hfill (not-pure)

Formula (not-pure) uses the less-than-or-equal predicate $\leq$ which is not allowed.\(^2\)

(a) Describe a pure equality formula that means that there are exactly two elements in the domain of discourse.

Solution. A simple answer using the above formulas is:

$$\text{formula (}\leq \text{2-elements)} \text{ AND NOT(formula (1-element)).}$$

An alternative is to use a variant of (≤ 2-elements):

$$\exists a, b. a \neq b \text{ AND } \forall x. x = a \text{ OR } x = b.$$  \hfill ■

(b) Describe a pure equality formula that means that there are exactly three elements in the domain of discourse.

---

\(^1\) Remember, a domain of discourse is not allowed to be empty.

\(^2\) In fact, formula (not-pure) only makes sense when the domain elements are ordered, while pure equality formulas make sense over every domain.
Solution. We can say there are at most three elements using

\[ \exists a, b, c \forall x. x = a \text{ OR } x = b \text{ OR } x = c. \quad (\leq 3\text{-elements}) \]

Then exactly three elements can be expressed with

formula \((\leq 3\text{-elements}) \text{ AND NOT} (\text{formula } (\leq 2\text{-elements}))\).

An alternative is

\[ \exists a, b, c. a \neq b \text{ AND } b \neq c \text{ AND } a \neq c \text{ AND } \forall x. x = a \text{ OR } x = b \text{ OR } x = c. \]