Faculty of Information Technology
Semester 2, 2018

FIT2014
Solutions for Tutorial 1
Logic and Proofs

1. We must show that, if $w \in$ PALINDROMES then $w \in \overline{\text{ODD-ODD}}$.

Suppose $w \in$ PALINDROMES. Then there is a string x such that *either* $w = x\overleftarrow{x}$ *or* $w = xy\overleftarrow{x}$, where $\overleftarrow{x}$ denotes the reverse of x and $y \in \{\text{a}, \text{b}\}$ is a single letter.

Suppose $w = x\overleftarrow{x}$. The numbers of a's and b's in $x\overleftarrow{x}$ are both even, since each is twice the number in x. So $w \in \overline{\text{ODD-ODD}}$.

Now suppose $w = xy\overleftarrow{x}$. Whichever letter in $\{\text{a}, \text{b}\}$ is *not* y must appear an even number of times in w, by the same argument we have used previously. So that letter does not appear an odd number of times in w. So $w \in \overline{\text{ODD-ODD}}$.

Alternative argument for the second case, pointed out by an FIT2014 student in 2013:

Now suppose $w = xy\overleftarrow{x}$. Since the length of w is odd, then it cannot have an both an odd number of a's and an odd number of b's (else its length would be even). So $w \in \overline{\text{ODD-ODD}}$.

2.

(a) k-th odd number $= 2k - 1$

(b) Inductive basis: when $k = 1$, the sum of the first k odd numbers is just the first odd number, 1, which equals 1^2, so it equals k^2.

(c)

$$\begin{aligned}
&\text{Sum of the first } k+1 \text{ odd numbers} \\
&\quad = \quad 1 + 3 + \cdots + ((k+1)\text{-th odd number}) \\
&\quad = \quad (1 + 3 + \cdots + (k\text{-th odd number})) + ((k+1)\text{-th odd number}) \\
&\quad = \quad (\text{sum of the first } k \text{ odd numbers}) + ((k+1)\text{-th odd number}) \\
&\quad = \quad (\text{sum of the first } k \text{ odd numbers}) + 2(k+1) - 1 \\
&\qquad\qquad\qquad (\text{using our formula from part (a)})
\end{aligned}$$

(d) Continuing from above,

$$\begin{aligned}
&\ldots \\
&\quad = \quad k^2 + 2(k+1) - 1 \qquad\qquad (\text{by the Inductive Hypothesis})
\end{aligned}$$

(e) Continuing from above,

$$\begin{aligned}
&\ldots \\
&\quad = \quad k^2 + 2k + 1 \\
&\quad = \quad (k+1)^2.
\end{aligned}$$

This completes the Inductive Step (i.e., going from k to $k+1$).

(f) So, by the Principle of Mathematical Induction, it is true for all k that the sum of the first k odd numbers is k^2.

3. Base case: $n = 1$:

The tree with one vertex has zero edges, which is $1-1$, so the claim is true for $n = 1$.

Inductive step:

Suppose $k > 1$, and that any tree with $k-1$ vertices has $(k-1)-1 = k-2$ edges.

Let T be any tree with k vertices.

Now, every tree with ≥ 2 vertices has a leaf, and removing any leaf from a tree gives another tree with one fewer vertex.

So, remove a leaf from T. Let T^- be the smaller tree so obtained. It has $k-1$ vertices, so we can apply the Inductive Hypothesis to it. This tells us that T^- has $k-2$ edges. Since we only deleted one edge when we deleted the leaf, this implies that T has $(k-2)+1$ edges, i.e., $k-1$ edges. This completes the inductive step.

Therefore, by Mathematical Induction, it is true that, for all n, every tree on n vertices has $n-1$ edges.

4.

Base case: $n = 3$:

$3! = 6$, while $(3-1)^3 = 2^3 = 8$, so the inequality is true for $n = 3$.

Inductive Step:

Suppose that $n! \leq (n-1)^n$ is true for a particular number n, where $n \geq 3$.

Let's look at what happens at $n+1$.

$$
\begin{aligned}
(n+1)! &= (n+1)\cdot n! \qquad \text{(to express it in terms of a smaller case)} \\
&\leq (n+1)\cdot(n-1)^n \qquad \text{(by the Inductive Hypothesis, i.e., } n! \leq (n-1)^n\text{)} \\
&= (n+1)(n-1)(n-1)^{n-1} \qquad \text{(a slight rearrangement} \ldots \\
&\qquad\qquad \ldots \text{we are hoping to get } n+1 \text{ factors, all} \leq n \ldots\text{)} \\
&= (n^2-1)(n-1)^{n-1} \qquad \text{(a little high-school algebra} \ldots\text{)} \\
&< n^2(n-1)^{n-1} \qquad \text{(and now we } \textit{do} \text{ have } n+1 \text{ factors, all} \leq n\text{)} \\
&< n^{n+1} \\
&= ((n+1)-1)^{n+1}.
\end{aligned}
$$

This last line is just to make it clear that the expression is of the required form.

So, by Mathematical Induction, it is true for all n that $n! \leq (n-1)^n$.

5.

1. $\mathbf{S_A} = \mathbf{B} \wedge \neg \mathbf{C}$
2. $\mathbf{S_B} = \mathbf{A} \rightarrow \mathbf{C}$
3. $\mathbf{S_C} = \neg \mathbf{C} \wedge (\mathbf{A} \vee \mathbf{B})$
4. Yes, since
$$(\mathbf{S_A} \wedge \mathbf{S_B} \wedge \mathbf{S_C}) \rightarrow (\mathbf{B} \wedge \neg \mathbf{A} \wedge \neg \mathbf{C})$$
is a tautology.

6.

(a) We prove it by constructing the truth table of each. It can be convenient to do this in stages.

P	Q	R	$Q \wedge R$	$P \vee (Q \wedge R)$
F	F	F	F	F
F	F	T	F	F
F	T	F	F	F
F	T	T	T	T
T	F	F	F	T
T	F	T	F	T
T	T	F	F	T
T	T	T	T	T

P	Q	R	$P \vee Q$	$P \vee R$	$(P \vee Q) \wedge (P \vee R)$
F	F	F	F	F	F
F	F	T	F	T	F
F	T	F	T	F	F
F	T	T	T	T	T
T	F	F	T	T	T
T	F	T	T	T	T
T	T	F	T	T	T
T	T	T	T	T	T

The right-hand columns of each table are identical, so the two expressions (at the tops of those columns) are logically equivalent.

(b)

$$\begin{aligned}
P \wedge (Q \vee R) &= \neg\neg P \wedge (\neg\neg Q \vee \neg\neg R) \\
&= \neg\neg P \wedge \neg(\neg Q \wedge \neg R) && \text{(by one of De Morgan's Laws)} \\
&= \neg(\neg P \vee (\neg Q \wedge \neg R)) && \text{(by the other of De Morgan's Laws)} \\
&= \neg((\neg P \vee \neg Q) \wedge (\neg P \vee \neg R)) && \text{(by part (a) of this question)} \\
&= \neg(\neg(P \wedge Q) \wedge \neg(P \wedge R)) && \text{(by De Morgan, twice)} \\
&= \neg\neg(P \wedge Q) \vee \neg\neg(P \wedge R) && \text{(by De Morgan, one last time)} \\
&= (P \wedge Q) \vee (P \wedge R)
\end{aligned}$$

Here we have used equality to stand for logical equivalence, which is normal.

7.

$$\begin{aligned}
(P_1 \wedge \cdots \wedge P_n) \Rightarrow C &= \neg(P_1 \wedge \cdots \wedge P_n) \vee C \\
&= (\neg P_1 \vee \cdots \vee \neg P_n) \vee C,
\end{aligned}$$

using De Morgan's Law.

Remark:

A disjunction of the form $\neg P_1 \vee \cdots \vee \neg P_n \vee C$, where $P_1, \ldots, P_n, C$ are each variables that can be True or False, is called a *Horn clause*. These play a big role in the theory of logic programming.

8.

(a) $L_{B,n} \vee L_{W,n} \vee L_{U,n}$

(b) If $L_{B,n}$ is true, then it's ok for vertex $n+1$ to be Black (since it then joins the black chain that includes vertex n, which must be ok as the position up to vertex n is legal). It could also be Uncoloured, since adding a new uncoloured vertex next to an existing vertex can never make a legal position illegal. But vertex $n+1$ cannot be White, as it is then in a chain of its own which has no Uncoloured neighbour.
Similarly, if $L_{W,n}$ is true, then vertex $n+1$ can be White or Uncoloured, but it cannot be Black.
Lastly, if $L_{U,n}$ is true, then vertex $n+1$ can be in any of the three states, since if it is coloured then it forms a chain of one vertex that already has an uncoloured neighbour, namely vertex n.

(c) If $A_{B,n}$ is true, then vertex $n+1$ can be Uncoloured, since that never hurts legality. But it cannot be Black or White. If it were Black, then it would join the Black chain that contains vertex n but does not yet have an uncoloured neighbour, so the position would remain almost legal but it wouldn't be legal. If vertex n were White, it would become a single-vertex chain with no uncoloured neighbour, so the position would be illegal. (Furthermore, the Black chain containing vertex n would not have an uncoloured neighbour, giving another reason for illegality, so the position is now not even *almost* legal.)
The same holds true for $A_{W,n}$: vertex $n+1$ can be Uncoloured, but not Black or White.
$A_{U,n}$ is impossible, since an almost legal position must have its final vertex coloured.

(d)
$L_{B,n+1}$ can be expressed as

$$(L_{B,n} \vee L_{U,n}) \wedge V_{B,n+1}.$$

$L_{W,n+1}$ can be expressed as

$$(L_{W,n} \vee L_{U,n}) \wedge V_{W,n+1}.$$

$L_{U,n+1}$ can be expressed as

$$(L_{B,n} \vee L_{W,n} \vee L_{U,n} \vee A_{B,n} \vee A_{W,n}) \wedge V_{U,n+1}.$$

$A_{B,n+1}$ can be expressed as

$$(L_{W,n} \vee A_{B,n}) \wedge V_{B,n+1}.$$

$A_{W,n+1}$ can be expressed as

$$(L_{B,n} \vee A_{W,n}) \wedge V_{W,n+1}.$$

9.

i. **taller(father(max), max)** $\wedge \neg$ **taller(father(max), father(claire))**

ii. $\exists X$ **taller(X, father(claire))**

iii. $\forall X \exists Y$ **taller(X, Y)**

iv. $\forall X$ **(taller(X, claire)** $\rightarrow$ **taller(X, max))**

Supplementary exercises

10. (a)

(b)

Inductive basis:

Suppose $n \leq 100$. Applying `wc` to standard output of this length gives a one-line standard output stating the numbers of lines, words and characters, with the number of characters being n. This output has 1 line, 3 (or 4?) words and some small number of characters consisting of the digit 1, the digit 3, a couple of digits (at most) for n, and some number of spaces (say, 21 altogether, but the analysis is much the same if this number is different). Applying `wc` again gives one line with these numbers in it: 1, 3, 25, again alongside 21 spaces. Another application of `wc` gives the same result. So the claim is true for $n \leq 99$.

Inductive step:

Now suppose the claim is true when the file/string has $\leq k$ characters, where $k \geq 100$. Suppose we are given a file/string of $k+1$ characters. Applying `wc` gives a one-line standard output, giving numbers of lines, words and characters as l, w and $k+1$, respectively, set out something like

$$l \qquad w \qquad k+1$$

For any number x, write digits(x) for the number of digits in x. Our one-line output has some number s of spaces, say $s = 21$; the number of non-space characters is

$$\text{digits}(l) + \text{digits}(w) + \text{digits}(k + 1).$$

Now, $l \leq k + 1$ and $w \leq k + 1$.[1] Therefore the number of characters in the one-line output above is $\leq s + 3 \cdot \text{digits}(k + 1)$. We claim this is $\leq k$ if k is large enough.

> To see this, first try $k = 100$. Then
>
> $$\begin{aligned} s + 3 \cdot \text{digits}(k + 1) &= 21 + 3 \cdot \text{digits}(100 + 1) \\ &= 21 + 3 \cdot \text{digits}(101) \\ &= 21 + 3 \cdot 3 \\ &= 30, \end{aligned}$$
>
> which is indeed $\leq k$. (Only minor changes are needed here if the actual value of s is not 21; it will not be *much* different from 21.) Now, whenever k increases by 1, digits($k+1$) either stays the same or increases by 1. But it only increases rarely, when $k + 1$ becomes 100, then when it becomes 1000, and so on. So, given that $s + 3 \cdot \text{digits}(k + 1) \leq k$ for $k = 100$ (and by a good margin), this inequality continues to hold for all higher k.[2]

Since this is now $\leq k$, the Inductive Hypothesis tells us that some further applications of `wc` will eventually give constant output.

Therefore the result follows for all n, by the Principle of Mathematical Induction.

11.

1. $\mathbf{S_I} = (\mathbf{K} \wedge \neg\mathbf{A}) \rightarrow \mathbf{I}$
2. $\mathbf{S_A} = (\mathbf{I} \wedge \mathbf{K}) \rightarrow \mathbf{A}$
3. $\mathbf{S_K} = \mathbf{K} \leftrightarrow ((\neg\mathbf{I} \wedge \neg\mathbf{A} \wedge \neg\mathbf{K}) \vee (\neg\mathbf{I} \wedge \mathbf{A} \wedge \mathbf{K}) \vee (\mathbf{I} \wedge \neg\mathbf{A} \wedge \mathbf{K}) \vee (\mathbf{I} \wedge \mathbf{A} \wedge \neg\mathbf{K}))$
4. No, since
   $$(\mathbf{S_I} \wedge \mathbf{S_A} \wedge \mathbf{S_K}) \rightarrow (\mathbf{A} \wedge \mathbf{K} \wedge \neg\mathbf{I})$$
   is not a tautology.
5. Yes, since
   $$(\mathbf{S_I} \wedge \neg\mathbf{S_A} \wedge \mathbf{S_K}) \rightarrow (\mathbf{I} \wedge \mathbf{K} \wedge \neg\mathbf{A})$$
   is a tautology.

---

[1] In fact, $w \leq (k+1)/2$, since every consecutive pair of words must have at least one space between them. But we don't need this better upper bound on w.

[2] Thanks to FIT2014 tutor Nathan Companez for part of this argument.