Monash University
Faculty of Information Technology
Semester 2, 2022

FIT2014
Solutions for Tutorial 2
Quantifiers, Games, Proofs

1.

(a)

$$\texttt{supervised}(\texttt{Alonzo Church}, \texttt{Alan Turing})$$

(b)

$$\forall X\ \neg\texttt{supervised}(X, X)$$

(c)

$$\exists X\ ((\exists Y \texttt{supervised}(X, Y)) \wedge \forall Y\ \neg\texttt{supervised}(Y, X))$$

(d)

$$\exists X\ (\texttt{supervised}(\texttt{Alan Turing}, X) \wedge \forall Y (\texttt{supervised}(\texttt{Alan Turing}, Y) \Rightarrow Y = X))$$

Alan Turing's sole PhD graduate was Robin Gandy (1919–1995), who became a significant figure in computability theory and finished his career at Oxford.

2.

(a) $\mathsf{CrossesToMove}(P)\ \wedge\ \exists X : \mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P, X)\,)$
or $\exists X : \mathsf{CrossesToMove}(P)\ \wedge\ \mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P, X)\,)$

(b) $\mathsf{NoughtsToMove}(P)\ \wedge\ \exists X : \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P, X)\,)$
or $\exists X : \mathsf{NoughtsToMove}(P)\ \wedge\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P, X)\,)$

(c) The statement (a) is False, because it is not Crosses' turn. The statement (b) is True, because Noughts can play in the top left cell to complete a line.

(d)

$$\begin{aligned}\forall P\ :\ (\ &(\mathsf{CrossesToMove}(P)\ \Rightarrow\ \forall X\,\neg\,\mathsf{NoughtsWins}(\mathsf{ResultingPosition}(P, X)))\\ &\wedge (\mathsf{NoughtsToMove}(P)\ \Rightarrow\ \forall X\,\neg\,\mathsf{CrossesWins}(\mathsf{ResultingPosition}(P, X)))\ )\end{aligned}$$

(e) $\mathsf{CrossesToMove}(P)\ \ \wedge\ \ \exists X\ \forall Y\ \exists Z : \mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P, X, Y, Z)\,)$

(f) $\neg\,\mathsf{CrossesToMove}(P)\ \ \vee\ \ \forall X\ \exists Y\ \forall Z : \neg\,\mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P, X, Y, Z)\,)$

(g)

$\exists X_1\,\forall X_2\,\exists X_3\,\forall X_4\,\exists X_5\,\forall X_6\,\exists X_7\,\forall X_8\,\exists X_9\ :\ \mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)$

(h)

$\forall X_1\,\exists X_2\,\forall X_3\,\exists X_4\,\forall X_5\,\exists X_6\,\forall X_7\,\exists X_8\,\forall X_9\ :\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)$

The following is also correct (why?):[1]

$\forall X_1\,\exists X_2\,\forall X_3\,\exists X_4\,\forall X_5\,\exists X_6\,\forall X_7\,\exists X_8\ :\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)\,)$

[1] Thanks to FIT2014 tutor Zhi Hao Tan for spotting an error in an earlier version of (h)&(i) and helping correct it.

(i)

$$
\begin{array}{rl}
& (\,\exists X_1\ \forall X_2\ \exists X_3\ \forall X_4\ \exists X_5\ \forall X_6\ \exists X_7\ \forall X_8\ \exists X_9\ :\ \neg\,\mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)\,) \\
\wedge & (\,\forall X_1\ \exists X_2\ \forall X_3\ \exists X_4\ \forall X_5\ \exists X_6\ \forall X_7\ \exists X_8\ \forall X_9\ :\ \neg\,\mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)\,)
\end{array}
$$

In fact, using some basic properties of the game, it's possible to show that the second part of the above solution is sufficient:[2]

$$\forall X_1\ \exists X_2\ \forall X_3\ \exists X_4\ \forall X_5\ \exists X_6\ \forall X_7\ \exists X_8\ \forall X_9\ :\ \neg\,\mathsf{CrossesWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)$$

Why is that?

(j)

$$\exists X_1\ \exists X_2\ \exists X_3\ \exists X_4\ \exists X_5\ \exists X_6\ \exists X_7\ \exists X_8\ \exists X_9\ :\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)\,)$$

In fact, if Noughts wins at all, then it must win on the eighth move (i.e., Noughts's fourth move), because the last move (X_9) is by Crosses and that move cannot undo a line of Noughts already formed. So an alternative solution is:[3]

$$\exists X_1\ \exists X_2\ \exists X_3\ \exists X_4\ \exists X_5\ \exists X_6\ \exists X_7\ \exists X_8\ :\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)\,)$$

Furthermore, for Noughts to win from P_0, it is sufficient to create a line of three Noughts while no line of three crosses is created, and also every line of three Noughts is part of a position which has three Crosses such that those Crosses are *not* in a line. Therefore, it is possible for Noughts to win if and only if it is possible for Noughts to win within six moves (i.e., three moves each). Therefore another correct solution is:

$$\exists X_1\ \exists X_2\ \exists X_3\ \exists X_4\ \exists X_5\ \exists X_6\ :\ \mathsf{NoughtsWins}(\,\mathsf{ResultingPosition}(P_0, X_1, X_2, X_3, X_4, X_5, X_6)\,)$$

The justification for six-move solution makes use of some more detailed properties of Noughts and Crosses. The justification for the eight-move solution only uses the property that it is never a *disadvantage* to move in this game. (In other words, it would never be advantageous to *pass*, if that were allowed.) The nine-move solution uses no specific properties of the game at all, other than that it finishes in at most nine moves. It was all we were looking for in this question, as the focus is on predicate logic, quantifiers, and the relationship between quantifiers and assertions about moves in games.

3. Assume, by way of contradiction, that there exists a nonempty hereditary language that does not contain the empty string. Let L be such a language.

Since L is nonempty, it contains at least one string, and therefore it contains a shortest string. Let x be a shortest string in L. Since L does not contain the empty string (by assumption), x cannot be empty. So it has length ≥ 1, and therefore contains some letters.

Since L is hereditary, some string x^- obtained from x by deleting one letter of x must also belong to L. But this gives a member of L which is shorter than the shortest possible string in L. This is a contradiction. So our assumption, that there exists a nonempty hereditary language that does not contain the empty string, must be wrong. Therefore every nonempty hereditary language contains the empty string.

[2] Thanks to FIT204 tutor Nathan Companez for spotting and correcting an error in an earlier version of this note to the solution of part (i).

[3] Thanks to FIT2014 tutor Nhan Bao Ho for this observation and the next one.

4.

(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)

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

(d) Continuing from above,

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

(e) Continuing from above,

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

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.

5. 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 $k \geq 1$, and that any tree with k vertices has $k - 1$ edges.

Let T be any tree with $k + 1$ vertices.

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

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

6.

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 $n! \le (n-1)^n$ is true for a particular number n, where $n \ge 3$.
Let's look at what happens at $n+1$.

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

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

So, by Mathematical Induction, it is true <u>for all $n \ge 3$</u> that $n! \le (n-1)^n$.

Supplementary exercises

7. Inductive basis ($n=1$): P_1 is just x_1, so if $x_1 = \mathrm{F}$, then P_1 is immediately False. So the statement holds in this case.

Inductive step:
Let $k \ge 1$. Assume that, if $x_1 = \mathrm{F}$, then P_k is False (the Inductive Hypothesis).
Now, if $x_1 = \mathrm{F}$, then we have

$$\begin{aligned}
P_{k+1} &= P_k \wedge x_{k+1} \qquad \text{(as noted in the question)} \\
&= \mathrm{F} \wedge x_{k+1} \qquad \text{(by the Inductive Hypothesis)} \\
&= \mathrm{F},
\end{aligned}$$

which completes the inductive step.
Therefore the statement holds for all n, by Mathematical Induction.

8. (a)
(b)
Inductive basis:
Suppose $n \le 100$. 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 $n \le 99$.
Inductive step:
Now suppose the claim is true when the file/string has $\le k$ characters, where $k \ge 100$. 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 $w \leq k+1$.[4] 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.[5]

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.

[4] 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.

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