1 Propositional Formulas

Question 1a.

By structural induction.

Base case. Let $f = x$, where x is a propositional variable. Then x is equivalent to itself, and does not contain the $\vee$ symbol.

Inductive step. Assume $f_1, f_2 \in G$, and suppose $P(f_1)$ and $P(f_2)$, i.e., there exist $f_1', f_2' \in G$ such that f_1' is logically equivalent to f_1, and f_2' is logically equivalent to f_2, and f_1' and f_2' do not contain the $\vee$ symbol.

We'll show$P(\neg f_1)$, $P((f_1 \implies f_2))$, and $P((f_1 \vee f_2))$.

- $\neg f_1'$ is a propositional formula equivalent to $\neg f_1$ that does not contain $\vee$.
- $f_1' \implies f_2'$ is a propositional formula equivalent to $f_1 \implies f_2$ that does not contain $\vee$.
- We claim that $\neg f_1' \implies f_2'$ is logically equivalent to $f_1 \vee f_2$. Indeed,

$$\begin{aligned} \neg f_1' \implies f_2' &\equiv \neg\neg f_1' \vee f_2' \\ &\equiv f_1' \wedge f_2' && (\neg\neg f \equiv f) \\ &\equiv f_1 \wedge f_2 && (IH). \end{aligned}$$

  Furthermore, since f_1' and f_2' don't contain $\vee$, neither does $\neg f_1' \implies f_2'$. This completes the induction.

2 Roommate Matching

Question 2a.

Define $G = (V, E)$ where V is the set of students, and for all $a, b \in V$, E contains the edge $\{a, b\}$. Define the weight function, $w : E \to \mathbb{R}_{\geq 0}$ to be a notion of compatibility given the survey responses. I.e. $w(\{a, b\})$ gives the compatibility score between a and b. For example we could define

$w(\{a, b\}) =$ overlap in sleep hours$+$(5 - difference in cleanliness score)$+$number of interests in common

Then, the problem of assigning roommate pairs such that pairs have similar living preferences and a high overlap in hobbies is equivalent to solving the matching problem where you're trying to maximize the weight amongst the matchings with maximum cardinality. Since weights defined in this way correspond to compatibility, a matching that maximizes weights among the matchings in which everyone is involved is the roommate pair assignment that maximizes compatibility.

Question 2b.

You can redefine your weight function so that that it gives higher weights to pairs that have a large diversity of hobbies. For example, if A is the set of interests for person a, and B is the set of interests for person b, then

$w(\{a, b\}) =$ overlap in sleep hours$+$(5 - difference in cleanliness score)$+|A \cup B|$

3 Recurrences

Question 3a.

Claim: $T(n) = \Theta(n)$. Note that $T(n) = \Omega(n)$ work since $T(n) = 2T(n/5) + T(2n/5) + n \geq n$ since T is positive. We just need to show $T(n) = O(n)$. By the substitution method.

$$\begin{aligned}
T(n) &= 2T(n/5) + T(2n/5) + n \\
&= 2cn/5 + c2n/5 + n \\
&= n(4c/5 + 1).
\end{aligned}$$

We need this to be at most cn, so pick $c \geq 5$

4 Induction

Question 4a.

The first inequality is incorrect. To get the $\leq$, we are allowed to replace the RHS with something larger. The inductive hypothesis applied to $k-1$ says that $f(k-1) \leq (k-1)^2$, which implies $-f(k+1) \geq -(k-1)^2$, so we are not justified in replacing $-f(k+1)$ with $-(k-1)^2$.

Question 4b.

We'll show $\forall n \in \mathbb{N}, n \geq 1 . f(n) = n^2 - 1$. Since for any $n \in \mathbb{N}$, $n^2 - 1 \leq n^2$ this is clearly sufficient.

By (complete) induction.

Base case. For the base case, $f(1) = 0 = 1^2 - 1$, and $f(2) = 3 = 2^2 - 1$. Thus, the base case holds.

Inductive step. Let $k \in \mathbb{N}$ with $k \geq 2$, and assume for all $i \in \mathbb{N}$ with $1 \leq i \leq k$, $f(i) = i^2 - 1$, we'll show $f(k+1) = (k+1)^2 - 1$

$$
\begin{aligned}
f(k+1) &= 2f(k) - f(k-1) + 2 \\
&= 2(k^2 - 1) - ((k-1)^2 - 1) + 2 && \text{(IH)} \\
&= 2k^2 - 2 - ((k^2 - 2k + 1) - 1) + 2 \\
&= 2k^2 - (k^2 - 2k) \\
&= k^2 + 2k \\
&= k^2 + 2k + 1 - 1 \\
&= (k+1)^2 - 1
\end{aligned}
$$

5 Sandwich

Question 5a.

Not injective: $f(\text{a}, \epsilon) = aa = f(\epsilon, aa)$.

Question 5b.

Surjective. Let $w \in \text{Strings}$. $w = f(\epsilon, w)$.

Question 5c.

By Structural Induction.

Base Case. Every string in the B has length 0 or 1. Thus, there it is impossible for more than one character to appear an odd number of times.

Inductive Step. For any $\sigma \in \text{Chars}$, and $\varphi \in \text{Strings}$ let $\text{Count}(\sigma, \varphi)$ be the number of times a character σ appears in φ.

Let $\alpha, \beta \in X$, and suppose for the inductive hypothesis that at most one character appears an odd number of times in α, and β. We'll now show that the same is true for $f(\alpha, \beta) = \alpha\beta\alpha$.

Let σ be any character. Then we have $\text{Count}(\sigma, \alpha\beta\alpha) = 2\text{Count}(\sigma, \alpha) + \text{Count}(\sigma, \beta)$. Note that this is odd if and only if $\text{Count}(\sigma, \beta)$ is odd. By the inductive hypothesis, $\text{Count}(\sigma, \beta)$ is odd for at most one character. Therefore, $\text{Count}(\sigma, \alpha\beta\alpha)$ is odd for at most one character.

Question 5d.

First we'll show $X \subseteq P$, namely, that $\forall x \in X . x \in P$. By structural induction.

Base Case. For the base case, we have that ϵ, and all length 1 strings are equal to their reverse, and hence in P.

Inductive Step. Let $\alpha, \beta \in X$, and suppose $\alpha, \beta \in P$, that is $\alpha = \text{Reverse}(\alpha)$, and $\beta = \text{Reverse}(\beta)$. We'll show that $f(\alpha, \beta) = \alpha\beta\alpha \in P$. We have $\text{Reverse}(\alpha\beta\alpha) = \text{Reverse}(\alpha)\text{Reverse}(\beta)\text{Reverse}(\alpha) = \alpha\beta\alpha$, Thus, $\alpha\beta\alpha \in P$.

Now, we'll show that $P \subseteq X$. By complete induction on the length of the string.

Base Case. We'll show all length 0 and length 1 palindromes are in X. The length 0 and length 1 palindromes are exactly ϵ and a for $a \in \text{Chars}$, which conveniently is the base set for X. Hence, X contains them all.

Inductive Step. Let $k \in \mathbb{N}$ with $k \geq 1$, and assume that all strings of length at most k in P, are also in X. Let $\alpha \in P$ be any string of length $k + 1$. Since $\alpha \in P$, we have $\alpha = \text{Reverse}(\alpha)$. Therefore, the first and last character must be equal, thus, we can write $\alpha = \sigma\alpha'\sigma$, where $\sigma = \alpha[0]$, and $\alpha' = \alpha[1 : -1]$ (using Python notation). Note that since $\alpha = \text{Reverse}(\alpha)$, we have $\alpha' = \text{Reverse}(\alpha')$, and hence $\alpha' \in P$. By the inductive hypothesis, since α' is a string of length $k - 1$, $\alpha' \in P$. From the base case, σ is also in P. Thus $f(\sigma, \alpha') = \sigma\alpha'\sigma \in P$, and of course, $\sigma\alpha'\sigma = \alpha$.

Question 5e.

Domain	**Codomain**	**Injective?**	**Surjective?**
$P \times P$	P	✗	✓
Chars $\times$ Strings	Strings	✓	✗
Chars $\cup \{\epsilon\} \times$ Strings	Strings	✗	✓
$P \times$ Chars $\cup \{\epsilon\}$	P	✓	✗

6 Extra Credit: NOR

Question 6a.

Sketch: The setup is the same as in problem 1.

Here is the inductive step.

- $\neg A \equiv A \text{ NOR } A$. If A is True, then $A \vee A$ is true, so $A \text{ NOR } A$ is false, if A is False, then $A \vee A$ is False, so $A \text{ NOR } A$ is true.
- $A \vee B \equiv \neg\neg(A \vee B) \equiv \neg A \text{ NOR } B \equiv (A \text{ NOR } B) \text{ NOR } (A \text{ NOR } B)$.
- $A \implies B \equiv (\neg A) \vee B \equiv (\neg A \text{ NOR } B) \text{ NOR } (\neg A \text{ NOR } B) \equiv ((A \text{ NOR } A) \text{ NOR } B) \text{ NOR } ((A \text{ NOR } A) \text{ NOR } B)$.