Mathematical Logic Homework 03

Qiu Yihang

Oct.21, 2022

1 The Binary Relation of Membership in Effectively Decidable Subset of $\mathbb{N}$ is Not Effectively Decidable

Proof. Assume B is effectively decidable.

Then exists an algorithm $\mathcal{B}$ for determining the membership of input (m, n) in B.

Since $A_0, A_1, ..., A_n, ...$ is a listing of all effectively decidable sets of $\mathbb{N}$, we know there exists algorithms $\mathcal{A}_0, \mathcal{A}_1, ..., \mathcal{A}_n, ...$ for determining the membership in $A_0, A_1, ..., A_n, ...$

We can construct a subset A^* of $\mathbb{N}$ as follows. For any number $n \in \mathbb{N}$,

$$\begin{cases} n \notin A^*, & \text{if algorithm } \mathcal{B} \text{ returns "YES" on input } (n, n) \\ n \in A^*, & \text{if algorithm } \mathcal{B} \text{ returns "NO" on input } (n, n) \end{cases}$$

Now we prove A^* is an effectively decidable. We can construct the following algorithm $\mathcal{A}^*$.

Algo. $\mathcal{A}^*$
begin
  on **Input** n;
  Find $\mathcal{A}_n$ in the listing $\mathcal{A}_0, \mathcal{A}_1, ..., \mathcal{A}_m, ...$ and run algorithm $\mathcal{A}_n$ on n;
  if *the result is "YES"* **then** **Output:** "NO" **else** **Output:** "YES";
end

$$n \in A^* \Leftrightarrow \mathcal{A}_n \text{ on input } n \text{ returns "NO"} \Leftrightarrow \mathcal{A}^* \text{ on input } n \text{ returns "YES".}$$

$$n \notin A^* \Leftrightarrow \mathcal{A}_n \text{ on input } n \text{ returns "YES"} \Leftrightarrow \mathcal{A}^* \text{ on input } n \text{ returns "NO".}$$

Thus, $\mathcal{A}^*$ determines the membership in A^*, i.e. A^* is an effectively decidable subset of $\mathbb{N}$.

Since $A_0, A_1, ..., A_n, ...$ is a listing of ***all*** effectively decidable subsets of $\mathbb{N}$, exists $k \in \mathbb{N}$ s.t. $A^* = A_k$. Consider the membership of k in A^*.

When $k \in A^*$, $k \in A^* \Rightarrow \mathcal{A}^*$ returns "YES" on $k \Rightarrow \mathcal{A}_k$ returns "NO" on $k \Rightarrow k \notin A_k \Rightarrow k \notin A^*$.

When $k \notin A^*$, $k \notin A^* \Rightarrow \mathcal{A}^*$ returns "NO" on $k \Rightarrow \mathcal{A}_k$ returns "YES" on $k \Rightarrow k \in A_k \Rightarrow k \in A^*$.

Contradiction.

Thus, B is **not** effectively decidable. ■

2 For any Wff $\boldsymbol{\alpha}$, $\boldsymbol{s(\alpha) = 1 + c(\alpha)}$

Proof. Proof by induction.

$S = \{\alpha \mid \alpha \text{ is a wff and } s(\alpha) = 1 + c(\alpha).\}$

Now we prove $S =$ the set of all wffs.

<u>**BASE CASE.**</u> $\alpha = A$, where A is a sentence symbol.

$c(\alpha) = 0, s(\alpha) = 1$. Thus, $s(\alpha) = 1 + c(\alpha)$ holds, i.e. $\alpha \in S$.

<u>**INDUCTIVE CASE.**</u>

(1) $\alpha = (\neg\beta)$. Obvious $s(\alpha) = s(\beta), c(\alpha) = c(\beta)$. Assume $\beta \in S$.

Then $s(\alpha) = s(\beta) = 1 + c(\beta) = 1 + c(\alpha)$. Thus, $\alpha \in S$. $\square$

(2) $\alpha = (\beta \wedge \gamma)$. Then $s(\alpha) = s(\beta) + s(\gamma), c(\alpha) = c(\beta) + c(\gamma) + 1$.

Assume $\beta, \gamma \in S$. Then $s(\alpha) = s(\beta) + s(\gamma) = c(\beta) + 1 + c(\gamma) + 1 = 1 + c(\alpha)$.

Thus, $\alpha \in S$. $\square$

(3) Similarly, for cases where $\alpha = (\beta \vee \gamma)$ or $\alpha = (\beta \rightarrow \gamma)$ or $\alpha = (\beta \leftrightarrow \gamma)$,

when $\beta, \gamma \in S$, $s(\alpha) = s(\beta) + s(\gamma) = 1 + c(\beta) + 1 + c(\gamma) = 1 + c(\alpha)$.

Thus, $\alpha \in S$. $\square$

Therefore, $S =$ the set of all wffs.

Thus, for any wff α, $s(\alpha) = 1 + c(\alpha)$, i.e. the number of occurrences of sentence symbols in α is 1 greater than the number of binary connectives in α. $\blacksquare$

3 Parsing Tree of a Wff

Solution. The resulting parse tree is as follows.

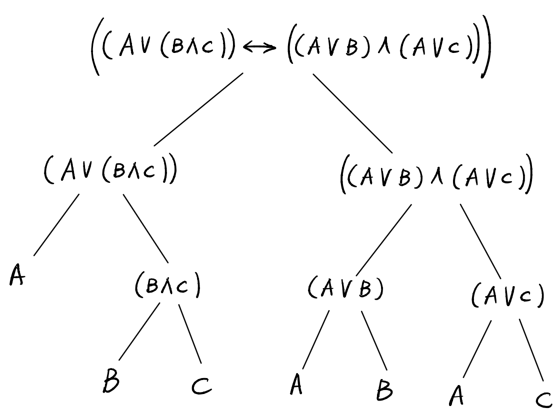

Figure 1: Parse Tree

How the algorithm constructs the parse tree above is explained as follows.

First create a single node $((A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C)))$.

The only existing leaf node is not a sentence symbol. Since the second symbol is not $\neg$, find $(A \vee (B \wedge C))$ followed by $\leftrightarrow$. The remaining part is $((A \vee B) \wedge (A \vee C))$. Create two children.

Leaf node $(A \vee (B \wedge C))$ is not a sentence symbol. Since the second symbol is not $\neg$, find

A

and $(B \wedge C)$. Create two children.

Leaf node $((A \vee B) \wedge (A \vee C))$ is not a sentence symbol. Since the 2nd symbol is not $\neg$, find $(A \vee B)$ and $(A \vee C)$. Create two children.

Leaf node A is a sentence symbol.

Leaf node $(B \wedge C)$ is not a sentence symbol. Since the 2nd symbol is not $\neg$, find B and C. Create two children.

Leaf node $(A \vee B)$ is not a sentence symbol. Since the 2nd symbol is not $\neg$, find A and B. Create two children.

Leaf node $(A \vee C)$ is not a sentence symbol. Since the 2nd symbol is not $\neg$, find A and C. Create two children.

All leaf nodes A, B, C, A, B, A, C are sentence symbols.

Terminate. ■

4 Problem 4

4.1 Whether $(\boldsymbol{P} \wedge \boldsymbol{Q}) \to \boldsymbol{R} \vDash (\boldsymbol{P} \to \boldsymbol{R}) \vee (\boldsymbol{Q} \to \boldsymbol{R})$ or not

Solution. $(P \wedge Q) \to R \vDash (P \to R) \vee (Q \to R)$. The proof is as follows.

Under truth assignment v,

$$\begin{aligned} & \bar{v}\big[(P \to R) \vee (Q \to R)\big] = \texttt{False} \\ \textbf{only if.} \quad & \bar{v}(P \to R) = \texttt{False} \text{ and } \bar{v}(Q \to R) = \texttt{False} \\ \textbf{only if.} \quad & v(P) = v(Q) = \texttt{True}, v(R) = \texttt{False}. \end{aligned}$$

v does not satisfy $((P \wedge Q) \to R)$, given that $\bar{v}\big[((P \wedge Q) \to R)\big] = \texttt{False}$.

Thus, truth assignments that do not satisfy $(P \to R) \vee (Q \to R)$

only if. they do not satisfy $((P \wedge Q) \to R)$,

i.e. truth assignments satisfying $((P \wedge Q) \to R)$ satisfies $(P \to R) \vee (Q \to R)$.

Therefore, <u>$(\boldsymbol{P} \wedge \boldsymbol{Q}) \to \boldsymbol{R}$ **tautologically implies** $(\boldsymbol{P} \to \boldsymbol{R}) \vee (\boldsymbol{Q} \to \boldsymbol{R})$</u>. ■

4.2 Whether $(P \wedge Q) \to R \models\!\!\!\!\dashv\ \models\!\!\dashv (P \to R) \vee (Q \to R)$ or not

Solution. $(P \wedge Q) \to R \models\!\!\dashv (P \to R) \vee (Q \to R)$. The proof is as follows.

The truth table of $(P \wedge Q) \to R$ and $(P \to R) \vee (Q \to R)$ is as follows.

$v(P)$	$v(Q)$	$v(R)$	$\bar{v}((P \wedge Q) \to R)$	$\bar{v}((P \to R) \vee (Q \to R))$
`True`	`True`	`True`	`True`	`True`
`True`	`True`	`False`	`False`	`False`
`True`	`False`	`True`	`True`	`True`
`True`	`False`	`False`	`True`	`True`
`False`	`True`	`True`	`True`	`True`
`False`	`True`	`False`	`True`	`True`
`False`	`False`	`True`	`True`	`True`
`False`	`False`	`False`	`True`	`True`

Thus, under any truth assignment v, $\bar{v}\big[(P \wedge Q) \to R\big] = \bar{v}\big[(P \to R) \vee (Q \to R)\big]$.

Therefore, any truth assignment satisfying $(P \wedge Q) \to R$ satisfies $(P \to R) \vee (Q \to R)$ while any truth assignment satisfying $(P \to R) \vee (Q \to R)$ satisfies $(P \wedge Q) \to R$.

Thus, **$(P \wedge Q) \to R$ is tautologically equivalent to $(P \to R) \vee (Q \to R)$.** ■

5 Problem 5

5.1 $((P \to Q) \to P) \to P$ Is a Tautology

Solution. $((P \to Q) \to P) \to P$ is a tautology. The proof is as follows.

The truth table of $((P \to Q) \to P) \to P$ is as follows.

$v(P)$	$v(Q)$	$\bar{v}[P \to Q]$	$\bar{v}\big[(P \to Q) \to P\big]$	$\bar{v}\big[((P \to Q) \to P) \to P\big]$
`True`	`True`	`True`	`True`	`True`
`True`	`False`	`False`	`True`	`True`
`False`	`True`	`True`	`False`	`True`
`False`	`False`	`True`	`False`	`True`

Thus, for any truth assignment v, $\bar{v}\big[((P \to Q) \to P) \to P\big] =$ `True`.

Therefore, $((P \to Q) \to P) \to P$ **is a tautology.** ■

5.2 $(\boldsymbol{A} \leftrightarrow \boldsymbol{B}) \rightarrow \neg((\boldsymbol{A} \rightarrow \boldsymbol{B}) \rightarrow \neg(\boldsymbol{B} \rightarrow \boldsymbol{A}))$ Is a Tautology

Solution. $(A \leftrightarrow B) \rightarrow \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))P$ is a tautology. The proof is as follows.

The truth table of $(A \leftrightarrow B) \rightarrow \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))$ is as follows.

$v(A)$	$v(B)$	$\bar{v}\big[(A \leftrightarrow B) \rightarrow \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))\big]$
`True`	`True`	`True`
`True`	`False`	`True`
`False`	`True`	`True`
`False`	`False`	`True`

Thus, for any truth assignment v, $\bar{v}\big[(A \leftrightarrow B) \rightarrow \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))\big] = \texttt{True}$.

Therefore, $(A \leftrightarrow B) \rightarrow \neg((A \rightarrow B) \rightarrow \neg(B \rightarrow A))$ **is a tautology.** ■

6 Problem 6

6.1 If $\boldsymbol{\Sigma} \models \boldsymbol{\alpha}$ Then For Any $\boldsymbol{\beta}$, $\boldsymbol{\Sigma} \models \boldsymbol{\beta} \rightarrow \boldsymbol{\alpha}$

Proof. $\Sigma \models \alpha \Longrightarrow$ for any truth assignment v satisfying Σ, $\bar{v}(\alpha) = \texttt{True}$.

$\Longrightarrow$ for any β, for any truth assignment v satisfying Σ, $\bar{v}(\beta \rightarrow \alpha) = \texttt{True}$.

(Since when $\bar{v}(\alpha) = \texttt{True}$, either $\bar{v}(\beta) = \texttt{True}$ or $\texttt{False}$, $\bar{v}(\beta \rightarrow \alpha) = \texttt{True}$.)

$\Longrightarrow$ for any β, for any truth assignment v satisfying Σ, v satisfies $\beta \rightarrow \alpha$.

$\Longrightarrow$ for any β, $\boldsymbol{\Sigma \models \beta \rightarrow \alpha}$. ■

6.2 $\boldsymbol{\Sigma}, \boldsymbol{\beta} \models \boldsymbol{\alpha}$ iff. $\boldsymbol{\Sigma} \models \boldsymbol{\beta} \rightarrow \boldsymbol{\alpha}$

Proof. First we prove $\Sigma, \beta \models \alpha \Longrightarrow \Sigma \models \beta \rightarrow \alpha$.

$\Sigma, \beta \models \alpha \Longrightarrow$ for any truth assignment v satisfying Σ and β, $\bar{v}(\alpha) = \texttt{True}$.

$\Longrightarrow$ for any truth assignment v satisfying Σ s.t. $\bar{v}(\beta) = \texttt{True}$, we have $\bar{v}(\alpha) = \texttt{True}$.

$\Longrightarrow$ for any truth assignment v satisfying Σ, $\bar{v}(\beta \rightarrow \alpha) = \texttt{True}$.

(Since for truth assignment v satisfying Σ,

When $\bar{v}(\beta) = \texttt{False}$, either $\bar{v}(\alpha) = \texttt{True}$ or $\texttt{False}$, $\bar{v}(\beta \rightarrow \alpha) = \texttt{True}$.

When $\bar{v}(\beta) = \texttt{True} \Longrightarrow \bar{v}(\alpha) = \texttt{True}$, we also have $\bar{v}(\beta \rightarrow \alpha) = \texttt{True}$.)

$\Longrightarrow$ for any truth assignment v satisfying Σ, v satisfies $\beta \rightarrow \alpha$.

i.e. $\Sigma \models \beta \rightarrow \alpha$. □

Now we prove $\Sigma \models \beta \to \alpha \Longrightarrow \Sigma, \beta \models \alpha$.

$\Sigma \models \beta \to \alpha \Longrightarrow$ for any truth assignment v satisfying Σ, $\bar{v}(\beta \to \alpha) = \texttt{True}$.

$\Longrightarrow$ for any truth assignment v satisfying Σ s.t. $\bar{v}(\beta) = \texttt{True}$, we have $\bar{v}(\alpha) = \texttt{True}$.

(Otherwise, $\bar{v}(\beta \to \alpha) = \texttt{False}$.)

$\Longrightarrow$ for any truth assignment v satisfying Σ and β, $\bar{v}(\alpha) = \texttt{True}$.

i.e. $\Sigma, \beta \models \alpha$. □

Therefore, $\Sigma \models \beta \to \alpha$ **iff.** $\Sigma, \beta \models \alpha$. ■