Solutions Manual

A Book of Abstract Algebra - 2nd Edition

Charles C. Pinter

This solution manual was created by the MathLearners study group.

Contents

Chapter 2

Operations

A: Examples of Operations

3 $a * b$ is a root of the equation $x^2 - a^2b^2 = 0$, on the set $\mathbb{R}$.

Solution. From $x^2 - a^2b^2 = 0$ we get $x^2 = a^2b^2$, so $\pm ab$ is a root, which means it's not unique. Thus $a * b$ is not an operation on $\mathbb{R}$.

5 Subtraction, on the set $\{n \in \mathbb{Z} | n \geq 0\}$.

Solution. Subtraction is not an operation on that set, because if we have $k, l \in \mathbb{Z}_{\geq 0}$, where $l > k$, we get a negative result, which would not be in $\mathbb{Z}_{\geq 0}$.

B: Properties of Operations

1 $x * y = x + 2y + 4$

Solution. We follow the steps from the example.

I. Commutative: $x * y = x + 2y + 4$; $y * x = y + 2x + 4$. Thus $x * y$ is not commutative.

II. Associative: $x * (y * z) = x * (y + 2z + 4) = x + 2(y + 2z + 4) + 4$
$(x * y) * z = (x + 2y + 4) * z = x + 2y + 4 + 2z$. Thus $x * y$ is not associative.

III. Solve $x * e = x$ for e.

$$\begin{aligned} x * e &= x \\ x + 2e + 4 &= x \\ 4 &= -2e \\ -2 &= e. \end{aligned}$$

IV. Solve $x * x' = e$ for x'.

$$\begin{aligned}
x * x' &= e \\
x + 2x' + 4 &= e \\
x + 2x' + 4 &= -2 \\
x + 2x' &= -6 \\
2x' &= -6 - x \\
x' &= -\frac{6+x}{2}.
\end{aligned}$$

2 $x * y = x + 2y - xy$

Solution. We follow the steps from the example.

I. Commutative:

$$\begin{aligned}
x * y &= x + 2y - xy \\
y * x &= y + 2x - yx.
\end{aligned}$$

Thus $x * y$ is not commutative.

II. Associative:

$$\begin{aligned}
x * (y * z) &= x * (y + 2z - yz) = x + 2(y + 2z - yz) - x(y + 2z - yz) \\
(x * y) * z &= (x + 2y - xy) * z = x + 2y - xy + 2z - (x + 2y - xy)z.
\end{aligned}$$

Thus $x * y$ is not associative.

III. Solve $x * e = x$ for e.

$$\begin{aligned}
x * e &= x \\
x + 2e - xe &= x \\
2e - xe &= 0 \\
e(2 - x) &= 0 \\
e &= 0.
\end{aligned}$$

Check that it works: $x * 0 = x + 2 \cdot 0 - x \cdot 0 = x + 0 - 0 = x$.

IV. Solve $x * x' = e$ for x'.

$$\begin{aligned}
x * x' &= e \\
x + 2x' - xx' &= 0 \\
x + x'(2 - x) &= 0 \\
x'(2 - x) &= -x \\
x' &= -\frac{x}{2-x}.
\end{aligned}$$

If $x = 2$, then the right side is undefined. Thus there is no inverse.

3 $x * y = |x + y|$

Solution. We follow the steps from the example.

I. Commutative:

$$\begin{aligned} x * y &= |x + y| \\ y * x &= |y + x| = |x + y|. \end{aligned}$$

Thus $x * y$ is commutative.

II. Associative:

$$\begin{aligned} x * (y * z) &= x * |y + z| = \big|x + |y + z|\big| \\ (x * y) * z &= |x + y| * z = \big||x + y| + z\big| \end{aligned}$$

Let $x = 2, y = -2$ and $z = 0$. Then $x * (y * z) = 2 * (-2 * 0) = \big|2 + |-2 + 0|\big| = |2 + 2| = 4$. But $(x * y) * z = (2 * -2) * 0 = \big||2 + (-2)| + 0\big| = |0 + 0| = 0$. Thus $x * y$ is not associative.

III. Solve $x * e = x$ for e.

$$\begin{aligned} x * e &= x \\ |x + e| &= x. \end{aligned}$$

But if $x < 0$, then the right side is negative, but the left side is nonnegative, which is a contradiction. Thus there is no identity element.

Chapter 3

The Definition of Groups

C: Groups of Subsets of a Set

1 Prove that there is an identity element with respect to the operation $+$, which is $\emptyset$.

Proof. Let A be any element of $\mathcal{P}(D)$. Then $A+\emptyset = (A-\emptyset)\cup(\emptyset-A) = A\cup\emptyset = A$. Thus $\emptyset$ is the identity element. ■

2 Prove every subset A of D has an inverse with respect to $+$, which is A.

Proof. Let A be any subset of D. Then $A + A = (A - A) \cup (A - A) = \emptyset \cup \emptyset = \emptyset$. Thus A is the inverse of A. ■

D: A Checkerboard Game

1 Write the table of G.

Solution. See table below:

$*$	V	H	D	I
V	I	D	H	V
H	D	I	V	H
D	H	V	I	D
I	V	H	D	I

2 Granting associativity, eplain why $\langle G, *\rangle$ is a group.

Solution. Assuming associativity, we only have to show that there exists an identity element and an inverse.

Proposition 1. The identity element of G is I.

Proof. Let X be any element of G. Then $X * I = X$ and $I * X = X$. Thus I is the identity element of G. ■

Proposition 2. For any element X of G, the inverse is X.

Proof. Let X be any element of G. Obersve that $X * X = I$ and $X * X = I$. Thus X is the inverse of X. ■

We also note that G is an abelian group, since changing the order of the operands doesn't change the result (i.e. $D * V = V * D = H$).

E: A Coin Game

1 If $G = \{I, M_1, ..., M_7\}$ and $*$ is the operation we have just defined, write the table of $\langle G, * \rangle$.

Solution. See table below:

I	I	M_1	M_2	M_3	M_4	M_5	M_6	M_7
I	I	M_1	M_2	M_3	M_4	M_5	M_6	M_7
M_1	M_1	I	M_3	M_2	M_5	M_4	M_7	M_6
M_2	M_2	M_3	I	M_1	M_6	M_7	M_4	M_5
M_3	M_3	M_2	M_1	I	M_7	M_6	M_5	M_4
M_4	M_4	M_6	M_5	M_7	I	M_2	M_1	M_3
M_5	M_5	M_7	M_4	M_6	M_1	M_3	I	M_2
M_6	M_6	M_4	M_7	M_5	M_2	I	M_3	M_1
M_7	M_7	M_5	M_6	M_4	M_3	M_1	M_2	I

2 Granting associativity, eplain why $\langle G, * \rangle$ is a group. Is it commutative? If not, show why not.

Solution. As can be seen from the operation table, there exists an identity element, namely I, and every element has an inverse. But G is not abelianc, since $M_4 * M_5 = M_2$, but $M_5 * M_4 = M_1$.

F: Groups in Binary Codes

1 Show that $(a_1, a_2, ..., a_n) + (b_1, b_2, ..., b_n) = (b_1, b_2, ..., b_n) + (a_1, a_2, ..., a_n)$.

Proof. We use induction and start with the base case:

I. $0 + 1 = 1 = 1 + 0$, and
$0 + 0 = 0 = 1 + 1$.

II. Assume that $(a_1, a_2, ..., a_k) + (b_1, b_2, ..., b_k) = (b_1, b_2, ..., b_k) + (a_1, a_2, ..., a_k)$ for $k \leq n$. Observe that $(a_1, a_2, ..., a_k, a_{k+1}) + (b_1, b_2, ..., b_k, b_{k+1}) = (b_1 + a_1, b_2 + a_2, ..., b_k + a_k, a_{k+1} + b_{k+1})$. Thus we have to show that $a_{k+1} + b_{k+1} = b_{k+1} + a_{k+1}$. Since a_{k+1} can be either 0 or 1, and b_{k+1} can also be either 0 or 1, we refer to the base case to conclude that $a_{k+1} + b_{k+1} = b_{k+1} + a_{k+1}$. Thus we have $(a_1, a_2, ..., a_k, a_{k+1}) + (b_1, b_2, ..., b_k, b_{k+1}) = (b_1, b_2, ..., b_k, b_{k+1}) + (a_1, a_2, ..., a_k, a_{k+1})$.

This completes our proof that $(a_1, a_2, ..., a_n) + (b_1, b_2, ..., b_n) = (b_1, b_2, ..., b_n) + (a_1, a_2, ..., a_n)$. ■

2 Check the remaining six cases:

Solution.

$$
\begin{aligned}
1 + (0 + 1) &= 1 + 1 = 0 = 1 + 1 = (1 + 0) + 1 \\
1 + (0 + 0) &= 1 + 0 = 1 = 1 + 0 = (1 + 0) + 0 \\
0 + (1 + 1) &= 0 + 0 = 0 = 1 + 1 = (0 + 1) + 1 \\
0 + (1 + 0) &= 0 + 1 = 1 = 1 + 0 = (0 + 1) + 0 \\
0 + (0 + 1) &= 0 + 1 = 1 = 0 + 1 = (0 + 0) + 1 \\
0 + (0 + 0) &= 0 + 0 = 0 = 0 + 0 = (0 + 0) + 0
\end{aligned}
$$

3 Show that $(a_1, ..., a_n) + [(b_1, ..., b_n) + (c_1, ..., c_n)] = [(a_1, ..., a_n) + (b_1, ..., b_n)] + (c_1, ..., c_n)$.

Proof. We use proof by induction and start with the base case.

I. See exercise 2.

II. Assume $(a_1, ..., a_k) + [(b_1, ..., b_k) + (c_1, ..., c_k)] = [(a_1, ..., a_k) + (b_1, ..., b_k)] + (c_1, ..., c_k)$ for $k \leq n$. Observe that the first k digits in $(a_1, ..., a_k, a_{k+1}) + [(b_1, ..., b_k, b_{k+1}) + (c_1, ..., c_k, c_{k+1})]$ are associative, which means whe only have to show that $a_{k+1} + (b_{k+1} + c_{k+1}) = (a_{k+1} + b_{k+1}) + c_{k+1}$ is true. And since we have shown in the base case that any binary word of length 1 is associative, we conclude that $a_{k+1} + (b_{k+1} + c_{k+1}) = (a_{k+1} + b_{k+1}) + c_{k+1}$ holds, and thus $(a_1, ..., a_k, a_{k+1}) + [(b_1, ..., b_k, b_{k+1}) + (c_1, ..., c_k, c_{k+1})] = [(a_1, ..., a_k, a_{k+1}) + (b_1, ..., b_k, b_{k+1})] + (c_1, ..., c_k, c_{k+1})$.

This completes our proof that addition of binary words is associative. ■

6 Show that $A + B = A - B$, [where $A - B = A + (-B)$].

Proof. Observe that:

$$\begin{aligned} A &= A \\ &= A + \mathbb{0} \\ &= A + (B + B) \\ &= (A + B) + B \\ A + (-B) &= (A + B) + B + (-B) \\ &= (A + B) + (B - B) \\ &= (A + B) + \mathbb{0} \\ A - B &= A + B. \end{aligned}$$

This completes our proof that $A + B = A - B$. ■

7 If $A + B = C$, show that $A = B + C$.

Proof. Observe that:

$$\begin{aligned} A + B &= C \\ (A + B) + B &= C + B \\ A + (B + B) &= \\ A + \mathbb{0} &= \\ A &= B + C. \end{aligned}$$

This completes our proof that $A + B = C$ implies $A = B + C$. ■

G: Theory of Coding: Maximum-Likelihood Decoding

The code, which we shall call $\mathcal{C}_1$, consists of the following binary words of length 5:

0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

1 Verify that every codeword $a_1a_2a_3a_4a_5$ in $\mathcal{C}_1$ satisfies the following two parity-check equations: $a_4 = a_1 + a_3$ and $a_5 = a_1 + a_2 + a_3$.

Solution. See the table below:

	a_4	a_5
000	$0+0=0$	$0+0+0=0+0=0$
001	$0+1=1$	$0+0+1=0+1=1$
010	$0+0=0$	$0+1+0=1+0=1$
011	$0+1=1$	$0+1+1=1+1=0$
100	$1+0=1$	$1+0+0=1+0=1$
101	$1+1=0$	$1+0+1=1+1=0$
110	$1+0=1$	$1+1+0=0+0=0$
111	$1+1=0$	$1+1+1=0+1=1$

2 Let $\mathcal{C}_2$ be the following code in $\mathbb{B}^6$. The first three positions are the information positions, and every codeword $a_1a_2a_3a_4a_5a_6$ satisfies the parity-check equations $a_4 = a_2$, $a_5 = a_1 + a_2$ and $a_6 = a_1 + a_2 + a_3$.

a. List the codewords of $\mathcal{C}_2$.

Solution. See the table below:

0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	1	1
0	1	1	1	1	0
1	0	0	0	1	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	0	1

b. Find the minimum distance of the code $\mathcal{C}_2$.

Solution. The minimum distance is 2.

c. How many errors in any codeword of $\mathcal{C}_2$ are sure to be detected? Explain.

Solution. We can be sure to detect at most one error in any codeword, since the minimum distance is 2. If we have 2 or more errors, the resulting binary word might be a codeword, in which case we can't determine that the received word contains an error, since it is valid.

3 Design a code in $\mathbb{B}^4$ where the first two positions are information positions. Give the parity-check equations, list the codewords, and find the minimum distance.

Solution. Let $a_3 = a_2$ and $a_4 = a_1 + a_2$. Then the code $\mathcal{C}_3$ has the following table:

0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

The minimum distance is 2.

4 Decode the following words in $\mathcal{C}_1$: 11111, 00101, 11000, 10011, 10001, and 10111.

Solution. The words are decoded as follows:

$$
\begin{aligned}
11111 &\rightarrow 11101 \\
00101 &\rightarrow 00111 \\
11000 &\rightarrow 11010 \\
10011 &\rightarrow 10011 \\
10001 &\rightarrow 10011 \\
10111 &\rightarrow 10011 \text{ or } 00111
\end{aligned}
$$

NOTE: Let $\mathcal{C}$ be a code in $\mathbb{B}^n$, m the minimum distance in $\mathcal{C}$, and A and B be codewords in $\mathcal{C}$.

5 Prove that it is possible to detect up to $m-1$ errors. (That is, if there are errors of transmission in $m-1$ or fewer positions of a codeword, it can always be determined that the received word is incorrect.)

Proof. We will use indirect proof. Thus assume we can't always determine that the received word is incorrect. This implies that the received word could be determined to be a correct one, which means the received word must be a codeword. Since we have a minimum distance of m between any two codewords, and only codewords are sent, there must have been errors in m or more positions. This completes our proof that we can always detect up to $m-1$ errors. ■

6 By the sphere of radius k about a codeword A we mean the set of all words in $\mathbb{B}^n$ whose distance from A is no greater than k. This set is denoted by $S_k(A)$; hence $S_k(A) = \{X \mid d(A,X) \leq k\}$. If $t = \frac{1}{2}(m-1)$, prove that any two spheres of radius t, say $S_t(A)$ and $S_t(B)$, have no elements in common.

Proof. We use proof by contradiction. Thus assume $t = \frac{1}{2}(m-1)$ and also assume that $S_t(A)$ and $S_t(B)$ have at least one element in common. Let W be that element. Thus $d(A,W) \leq \frac{1}{2}(m-1)$ and $d(B,W) \leq \frac{1}{2}(m-1)$, but also $d(A,B) \geq m$. This implies that $d(A,W) + d(B,W) \leq \frac{1}{2}(m-1) + \frac{1}{2}(m-1) = \frac{2}{2}(m-1) = m-1$. But $d(A,B) \leq d(A,W) + d(B,W)$, so $m \leq d(A,B) \leq m-1$, which is a contradiction. This completes our proof. ■

In our last proof, we assumed that $d(A,B) \geq d(A,W)+d(B,W)$ for all A,B,W, which could be called the triangle inequality of binary words. We will now prove this inequality.

Proof. Let A, B, W be any three binary words of length n. Let A and B differ in t positions and match in s positions. Then W can at best match A and B in s positions and has to differ from A or B in t positions. In this case $d(A, B) = d(A, W) + d(B, W)$. In all other cases, W will differ from A or B in $t + 1$ or more positions, which implies $d(A, B) < d(A, W) + d(B, W)$. Thus $d(A, B) \leq d(A, W) + d(B, W)$, which completes our proof. ∎