Introduction to Multiplication via Recursion

1 Definition

Multiplication of two non-negative integers a and b, denoted $a \times b$ or $a \cdot b$, can be defined recursively using addition:

- **Base Case**: If $b = 0$, then $a \times b = 0$.
- **Recursive Case**: If $b > 0$, then $a \times b = a + (a \times (b - 1))$.

This definition reduces multiplication to repeated addition. Each recursive step decrements b until it reaches the base case $b = 0$.

2 Examples

2.1 Example 1: 3×0

Applying the base case directly:

$$3 \times 0 = 0$$

2.2 Example 2: 0×4

Using the recursive definition:

$$\begin{aligned}
0 \times 4 &= 0 + (0 \times 3) \\
&= 0 + (0 + (0 \times 2)) \\
&= 0 + (0 + (0 + (0 \times 1))) \\
&= 0 + (0 + (0 + (0 + (0 \times 0)))) \\
&= 0 + (0 + (0 + (0 + 0))) \\
&= 0 + (0 + (0 + 0)) \\
&= 0 + (0 + 0) \\
&= 0 + 0 \\
&= 0
\end{aligned}$$

2.3 Example 3: 2×3

Breaking down the recursion step-by-step:

$$
\begin{aligned}
2 \times 3 &= 2 + (2 \times 2) \\
&= 2 + (2 + (2 \times 1)) \\
&= 2 + (2 + (2 + (2 \times 0))) \\
&= 2 + (2 + (2 + 0)) \\
&= 2 + (2 + 2) \\
&= 2 + 4 \\
&= 6
\end{aligned}
$$

2.4 Example 4: 5×2

Applying the recursive steps:

$$
\begin{aligned}
5 \times 2 &= 5 + (5 \times 1) \\
&= 5 + (5 + (5 \times 0)) \\
&= 5 + (5 + 0) \\
&= 5 + 5 \\
&= 10
\end{aligned}
$$

2.5 Example 5: 1×5

Demonstrating recursion with identity multiplication:

$$
\begin{aligned}
1 \times 5 &= 1 + (1 \times 4) \\
&= 1 + (1 + (1 \times 3)) \\
&= 1 + (1 + (1 + (1 \times 2))) \\
&= 1 + (1 + (1 + (1 + (1 \times 1)))) \\
&= 1 + (1 + (1 + (1 + (1 + (1 \times 0))))) \\
&= 1 + (1 + (1 + (1 + (1 + 0)))) \\
&= 1 + (1 + (1 + (1 + 1))) \\
&= 1 + (1 + (1 + 2)) \\
&= 1 + (1 + 3) \\
&= 1 + 4 \\
&= 5
\end{aligned}
$$

3 Conclusion

This recursive framework demonstrates how multiplication is equivalent to repeated addition. By systematically reducing b and leveraging the base case

$b = 0$, the definition breaks down complex operations into simpler, foundational steps. Recursion provides a clear algorithmic structure for understanding multiplication.