

7.2 情報科学の基礎理論

- 数の表現
- 集合論

目次

1 数の表現

2 集合論

数の表現

- コンピュータの基本は **0** と **1** で計算している。電源が Off の場合は 0、電源が On の場合は 1 となる。この電源 On と Off の二つの状態から計算を行う場合、最も簡単な表現方法は **2 進数**^[Binary] となる。

10進数

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12

10 個の数で表現

11 個目は桁上がり

2進数

0 → 1 → 10 → 11 → 100 → 101 → 110 → 111

2 個の数で表現

3 個目は桁上がり

基数変換

- 2 進数と 10 進数^[Decimal]の変換、または 8 進数や 16 進数といった基数を変換することを**基数変換**^[Base Conversion]と呼ぶ。

〈2 進数→10 進数の変換〉

2 進数:

1

1

0

1



掛ける

重み:

2^3

2^2

2^1

2^0

10 進数:

1×2^3

+

1×2^2

+

0×2^1

+

1×2^0

=

8

+

4

+

0

+

1

=

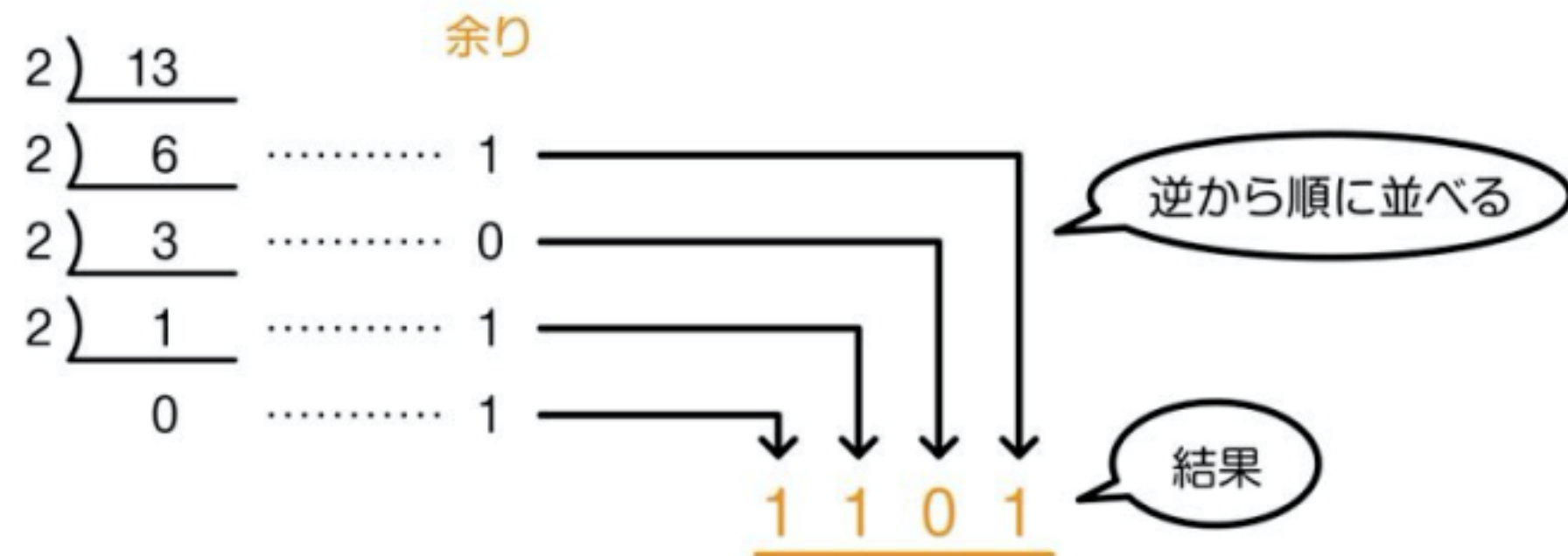
13

結果

次へ



〈10進数→2進数の変換〉



8 進数と 16 進数

- 2 進数と同じように、8・16 個の数で表現する方法を **8 進数**^[Octal]・**16 進数**^[Hexadecimal, Hex]という。

〈8 進数、16 進数→10 進数の変換〉

8 進数: 5 1 2

↑ ↑ ↑

重み: 8^2 8^1 8^0 ← 基数は 8

10 進数: $5 \times 64 + 1 \times 8 + 2 \times 1 = \underline{330}$

結果

16 進数: 1 F B

↑ ↑ ↑

重み: 16^2 16^1 16^0 ← 基数は 16

10 進数: $1 \times 256 + \frac{15}{(F)} \times 16 + \frac{11}{(B)} \times 1 = \underline{507}$

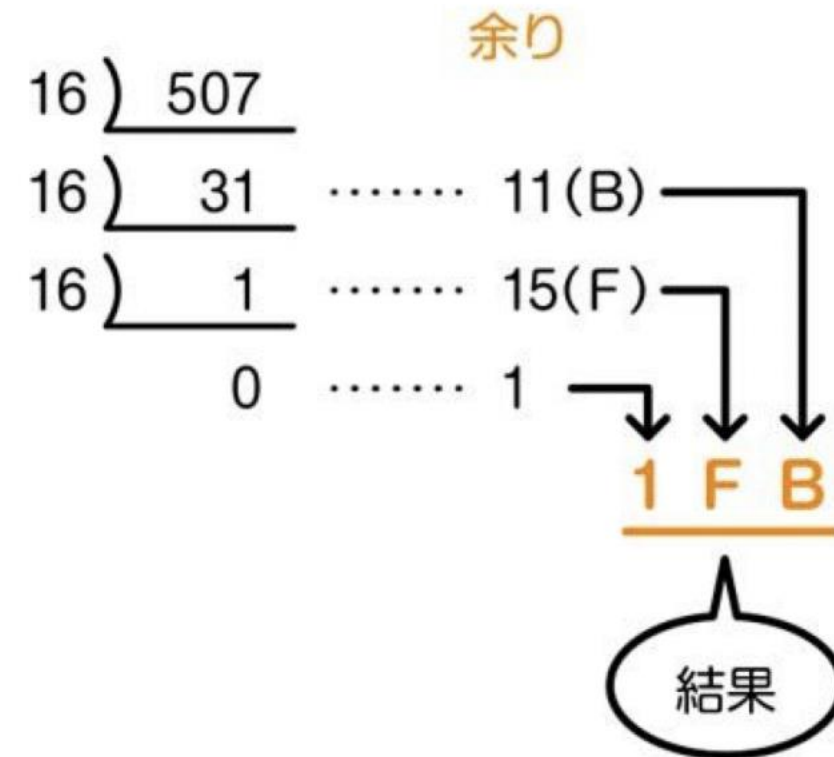
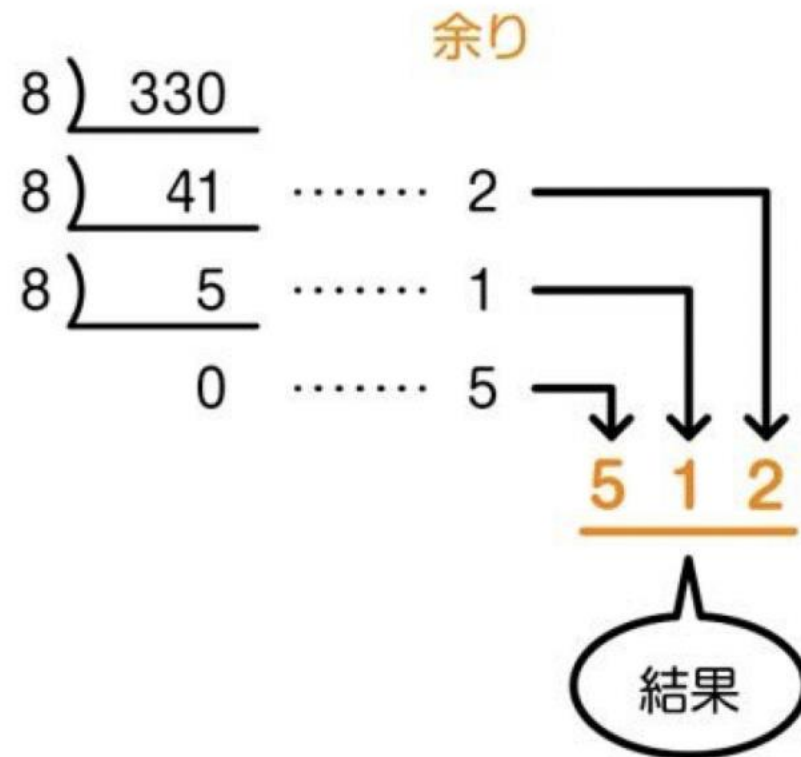
結果

次へ





〈10進数→8進数、16進数の変換〉



Note

16 進数の “数字” は 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F である。

2 進数の演算

- 2 進数の桁上がり : $1 + 1 = 10$ 。
- 2 進数の桁下がり : $10 - 1 = 1$ 。

$$\begin{array}{r}
 1011 \\
 + 0101 \\
 \hline
 10000 \\
 \\
 1011 \\
 - 0101 \\
 \hline
 110
 \end{array}$$

$$\begin{array}{r}
 1011 \\
 101 \\
 \hline
 1011 \\
 0000 \\
 1011 \\
 \hline
 110111
 \end{array}$$

進数対応

- 問題：人の指で最大何個の数を表現できるか考えてみよう！

Tips

16進数を表示する場合の表示方法：

1. 右下に小さく「16」と書く：17FA₁₆。
2. 冒頭に「0x」を書く
0x17FA。

10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

データ容量の単位

一度に最大で 32 ビットのデータを処理できる CPU を **32 ビット CPU**、64 ビットのデータを処理できる CPU を **64 ビット CPU** と呼ぶ。

● ビットとバイト

コンピュータは 2 進数ですべてのデータが構成される。この時の最小単位「2 進数の 1 桁分」を**ビット**[Bit]と呼び、**8 ビット**分をまとめた単位を**バイト**[Byte]と呼ぶ。



1 ビット (bit)



1 バイト (byte)

8 ビット (bit) = 1 バイト (byte)

● 補助単位

大きい値の補助単位		小さい値の補助単位	
k(キロ)	$10^3 = 1,000$ 倍	m(ミリ)	$1/10^3 = 1/1,000$
M(メガ)	$10^6 = 1,000,000$ 倍	μ(マイクロ)	$1/10^6 = 1/1,000,000$
G(ギガ)	$10^9 = 1,000,000,000$ 倍	n(ナノ)	$1/10^9 = 1/1,000,000,000$
T(テラ)	$10^{12} = 1,000,000,000,000$ 倍	p(ピコ)	$1/10^{12} = 1/1,000,000,000,000$

演習問題

- 【問題 1】

2進数1011と2進数101を乗算した結果の2進数はどれか。

平成28年秋期 問91

4問目／選択範囲の問題数51問

ア 1111

イ 10000

ウ 101111

エ 110111

演習問題

- 【問題 2】

10進数155を2進数で表したものはどれか。

出典：令和2年秋期 問62

ア

10011011

イ

10110011

ウ

11001101

エ

11011001

Q&A

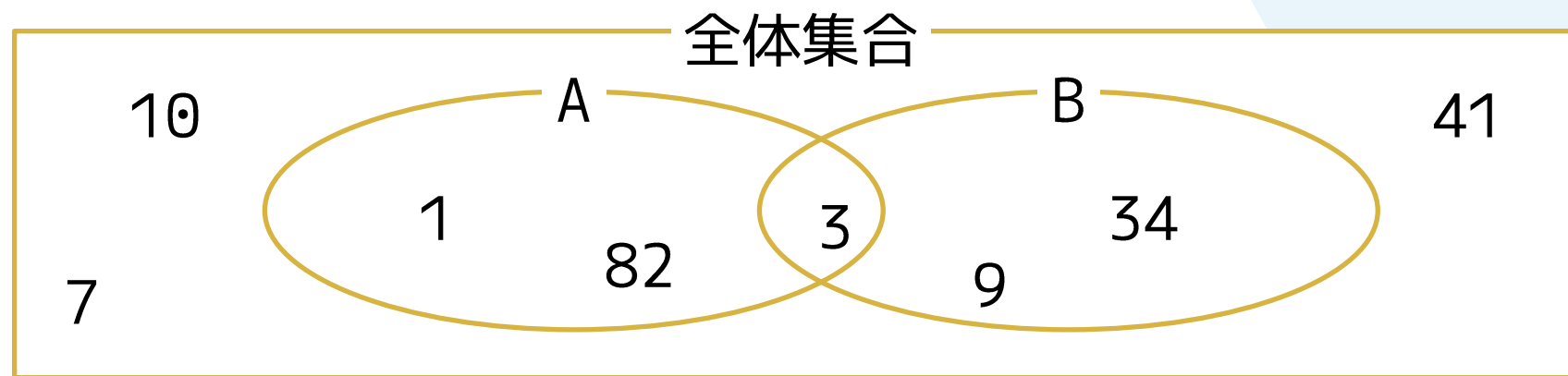
目次

1 数の表現

2 集合論

集合とは

- ある特性をもったデータ（要素）の集まりを**集合**^[Set]という。集合を表す図に**ベン図**^[Venn Diagram]がある。

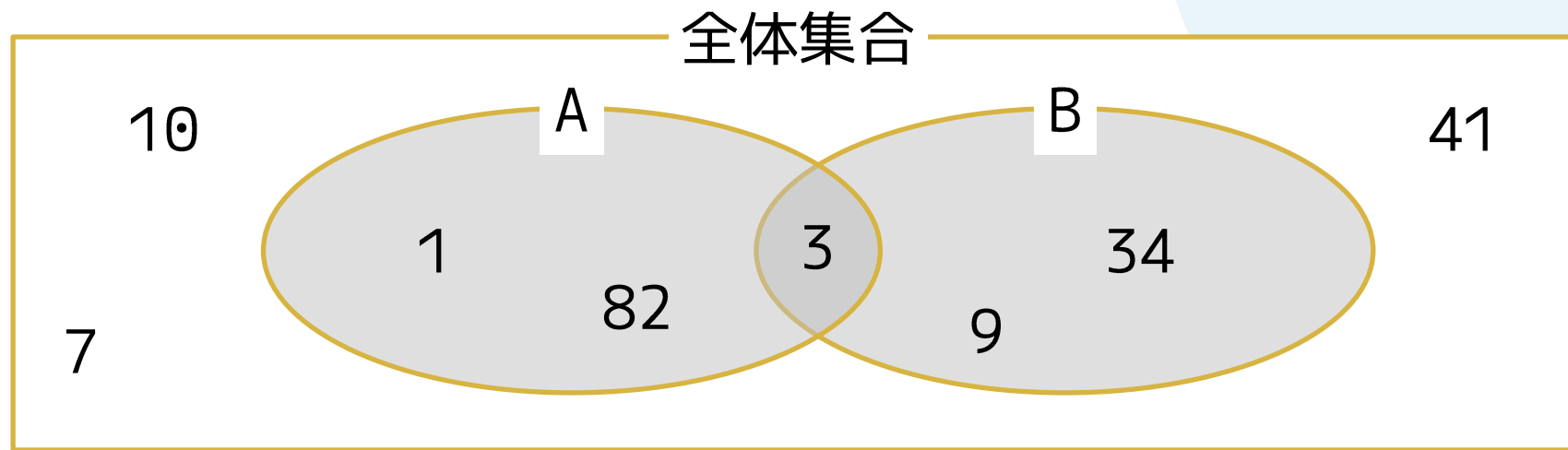


全体集合	対象とするデータすべてで構成される集合。
補集合	ある集合Aに対して「Aでない」要素の集まり。
部分集合	ある集合Aに対して「Aに含まれる」集合Bのこと。

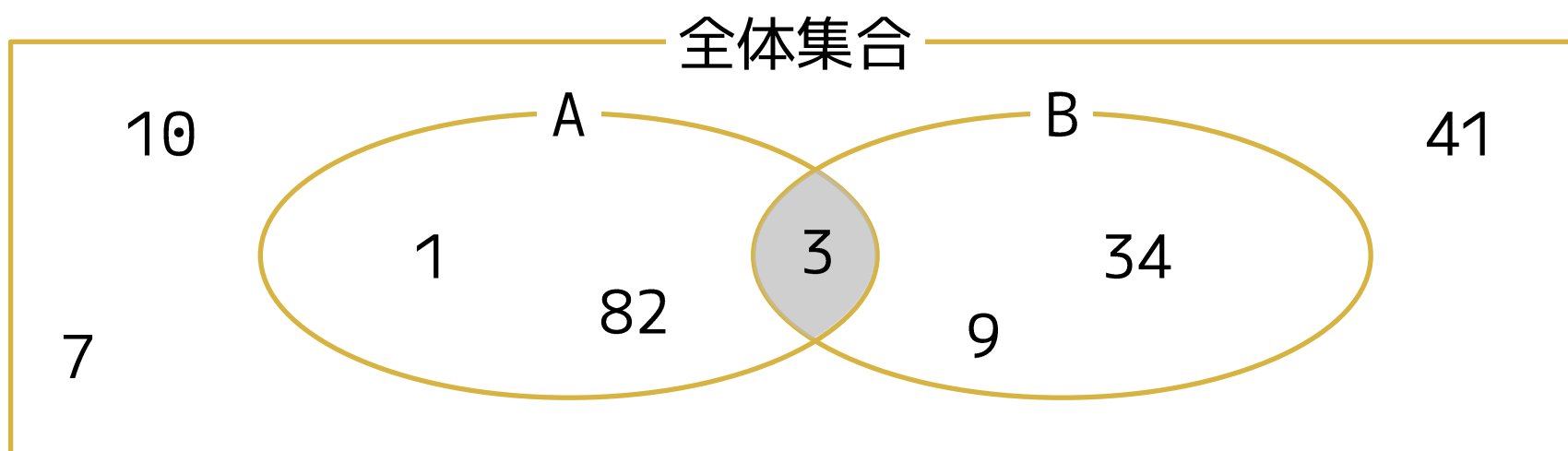
- コンピュータはデータを扱う時、条件で対象を絞り込みする。この絞り込み条件は集合論の概念を用いている。**かつ**^[AND]、**または**^[OR]などを見ていこう。

和集合と積集合

- **和集合**_[Union] : A または B (A **OR** B) 。

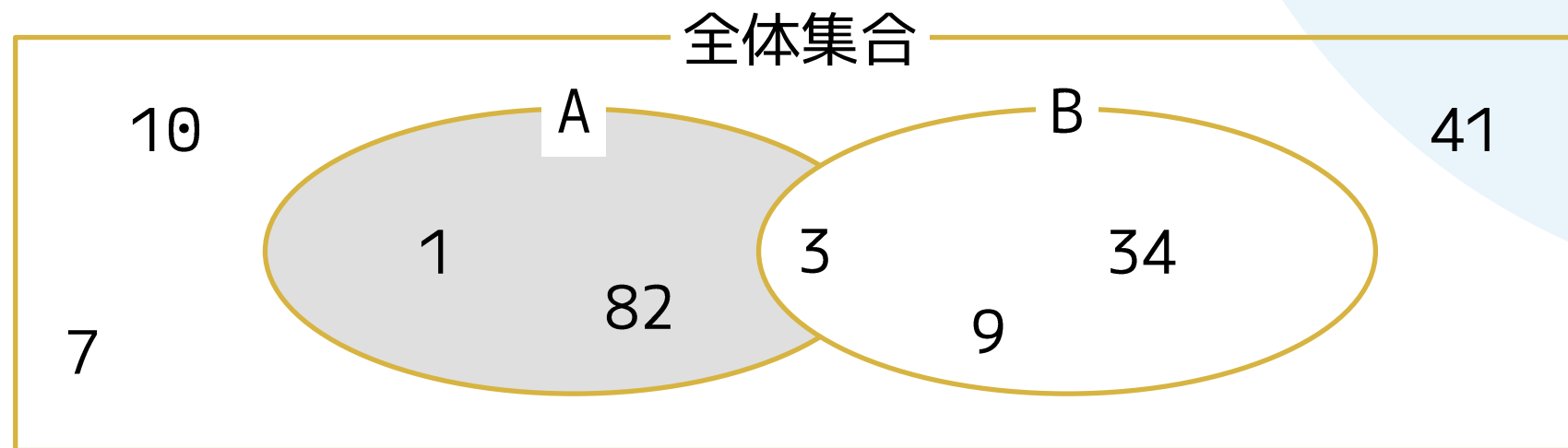


- **積集合**_[Product] : A **かつ** B (A **AND** B) 。



差集合

- 差集合^[Difference] : A かつ B でない (A AND **NOT** B) 。



集合演算と論理演算

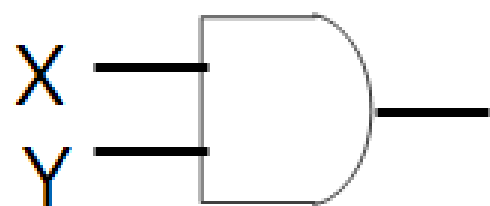
論理和 (OR、または)	二つの値がいずれも偽 (0) のときのみ結果が偽 (0) となり、それ以外は真 (1) となる
論理積 (AND、かつ)	二つの値がいずれも真 (1) のときのみ結果が真 (1) となり、それ以外は偽 (0) となる
否定 (NOT、ではない)	値が真 (1) のときに結果が偽 (0) となり、値が偽 (0) のときに結果が真 (1) となる
排他的論理和 (XOR)	二つの値が異なるときに結果が真 (1) となり、二つの値が等しいときに結果が偽 (0) となる

- **真**^[True]と**偽**^[False]：集合において、条件 (A) を満たすこと、集合 A に含まれることを「**真**」といい、条件 (A) を満たさない、集合 A に含まれないことを「**偽**」という。論理演算の場合、真を **1** で表し、偽を **0** で表すこともある。

論理積 (AND)

- 論理積は入力される二つの値 (X,Y) のどちらも「1」だった場合に、結果が「1」になる論理演算です。
- 論理回路は図にした時に一目で分かり易いように記号を使って表現されています。この記号のことを「**MIL記号 (ミル)**」と呼びます。

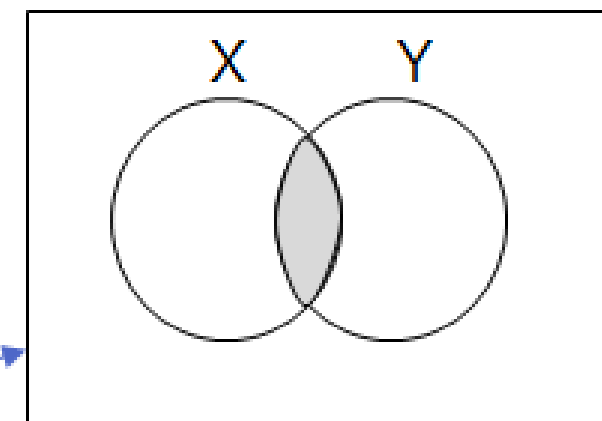
MIL記号
(AND回路)



真理値表 (AND)

X	Y	X・Y
0	0	0
0	1	0
1	0	0
1	1	1

ベン図 (AND)

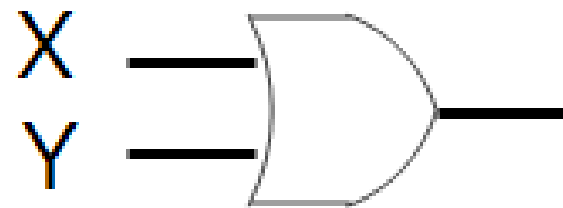


引用文献:<https://itmanabi.com/logical-operation/>

論理和 (OR)

- 論理和は入力される二つの値 (X,Y) のどちらかが「1」ならば、結果 (答え) が「1」になる論理演算です。

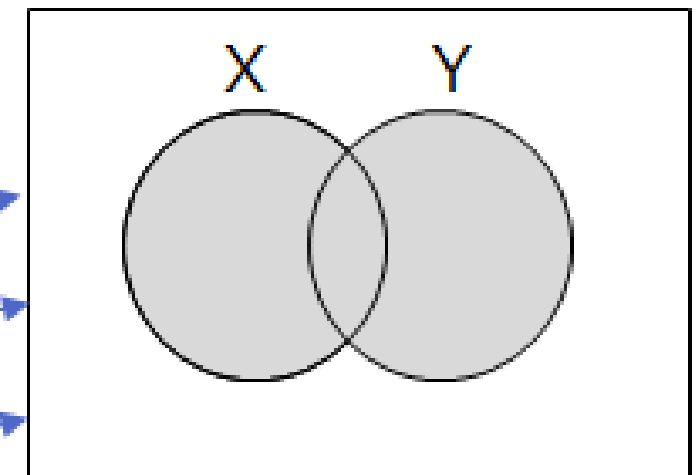
MIL記号
(OR回路)



真理値表 (OR)

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

ベン図 (OR)

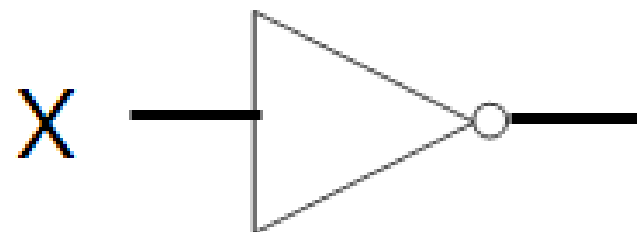


引用文献:<https://itmanabi.com/logical-operation/>

否定 (NOT)

- 否定は入力Xが「0」の場合、結果が反対の「1」になります。反対に入力Xが「1」であれば、結果が「0」になる論理演算です。

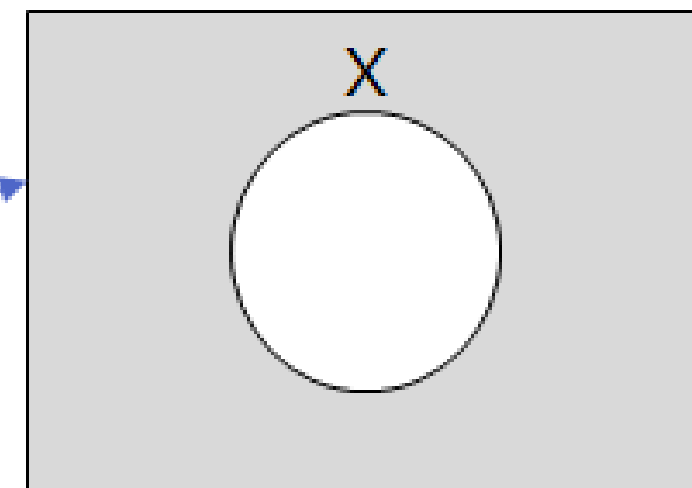
MIL記号
(NOT回路)



真理値表
(NOT)

X	\overline{X}
0	1
1	0

ベン図 (NOT)

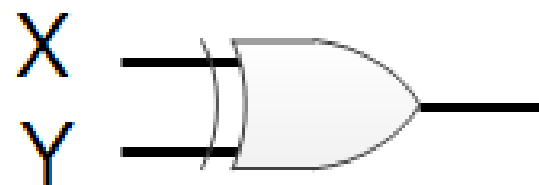


引用文献:<https://itmanabi.com/logical-operation/>

排他的論理和 (XOR)

- 排他的論理和は入力のXとYが異なる時に結果が「1」になり、同じとき（1と1か0と0）の時に結果が「0」になる論理演算です。

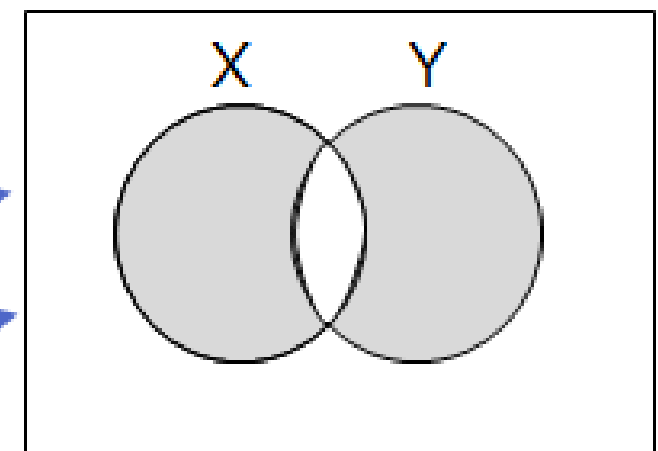
MIL記号
(XOR回路)



真理値表 (XOR)

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

ベン図 (XOR)



引用文献:<https://itmanabi.com/logical-operation/>

論理演算の法則

四則演算の交換法則

$$\begin{aligned} a + b &= b + a \\ a \times b &= b \times a \end{aligned}$$

四則演算の分配法則

$$a \times (b + c) = (a \times b) + (a \times c)$$

四則演算の結合法則

$$\begin{aligned} a + (b + c) &= (a + b) + c \\ a \times (b \times c) &= (a \times b) \times c \end{aligned}$$

論理演算の交換法則

$$\begin{aligned} a \text{ AND } b &= b \text{ AND } a \\ a \text{ OR } b &= b \text{ OR } a \end{aligned}$$

論理演算の分配法則

$$\begin{aligned} a \text{ AND } (b \text{ OR } c) &= (a \text{ AND } b) \text{ OR } (a \text{ AND } c) \\ a \text{ OR } (b \text{ AND } c) &= (a \text{ OR } b) \text{ AND } (a \text{ OR } c) \end{aligned}$$

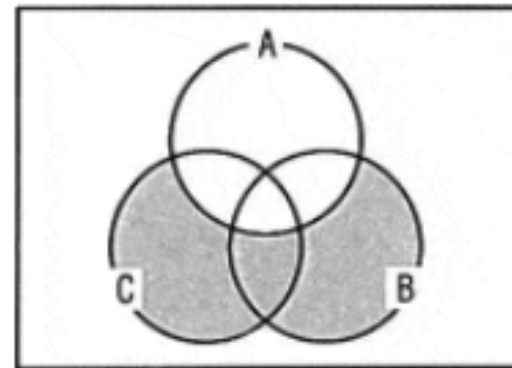
論理演算の結合法則

$$\begin{aligned} a \text{ AND } (b \text{ AND } c) &= (a \text{ AND } b) \text{ AND } c \\ a \text{ OR } (b \text{ OR } c) &= (a \text{ OR } b) \text{ OR } c \end{aligned}$$

演習問題

● 【問題 1】

次のベン図の網掛けした部分の検索条件はどれか。



平成29年秋期 問98

40問目／選択範囲の問題数51問

ア (not A) and (B and C)

イ (not A) and (B or C)

ウ (not A) or (B and C)

エ (not A) or (B or C)

演習問題

● 【問題 2】

二つの集合AとBについて、常に成立する関係を記述したものはどれか。ここで、 $(X \cap Y)$ は、XとYの両方に属する部分(積集合)、 $(X \cup Y)$ は、X又はYの少なくとも一方に属する部分(和集合)を表す。

平成27年春期 問62

24問目／選択範囲の問題数51問

ア $(A \cup B)$ は、 $(A \cap B)$ でない集合の部分集合である。

イ $(A \cup B)$ は、Aの部分集合である。

ウ $(A \cap B)$ は、 $(A \cup B)$ の部分集合である。

エ $(A \cap B)$ は、Aでない集合の部分集合である。

演習問題

● 【問題 3】

二つの集合AとBについて、常に成立する関係を記述したものはどれか。ここで、 $(X \cap Y)$ は、XとYの共通部分(積集合)、 $(X \cup Y)$ は、X又はYの少なくとも一方に属する部分(和集合)を表わす。

出典：平成22年春期 問69

ア $(A \cap B)$ は、Aでない集合の部分集合である。

イ $(A \cap B)$ は、Aの部分集合である。

ウ $(A \cup B)$ は、 $(A \cap B)$ の部分集合である。

エ $(A \cup B)$ は、Aの部分集合である。

Q&A

まとめ

Sum Up



1. 数の表現：

- ① 2 進数、8 進数、16 進数と 10 進数の変換。
- ② 2 進数の演算。
- ③ データ容量の単位。

2. 集合論：

- ① 集合の概念。
- ② 集合演算。
- ③ 論理演算。



Light in Your Career.
THANK YOU!