

任意の論理演算とその合成

前の節では \Rightarrow

\wedge (AND)、 \vee (OR)、 \neg (NOT)

の動作を理解する

この節では \Rightarrow

$\wedge \cdot \vee \cdot \neg$ を組合わせて

もっと複雑な関数を作る

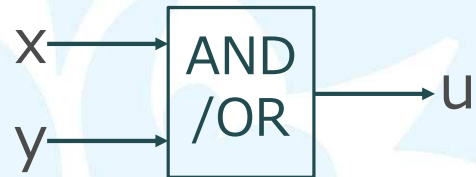
もっと複雑な論理関数とは

複雑な論理演算

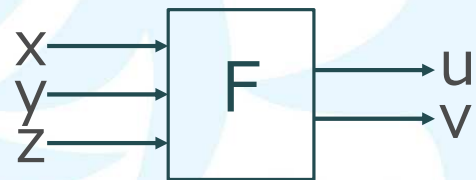
入出力が多い

たとえば

$$u = (x \wedge y) \wedge y \vee (y \vee x)$$



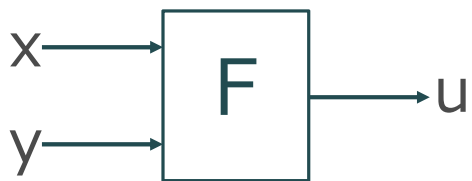
3入力2出力だと



東邦大学

2

複雑な論理演算



たとえば

$$u = (x \wedge y) \wedge y \vee (y \vee x)$$

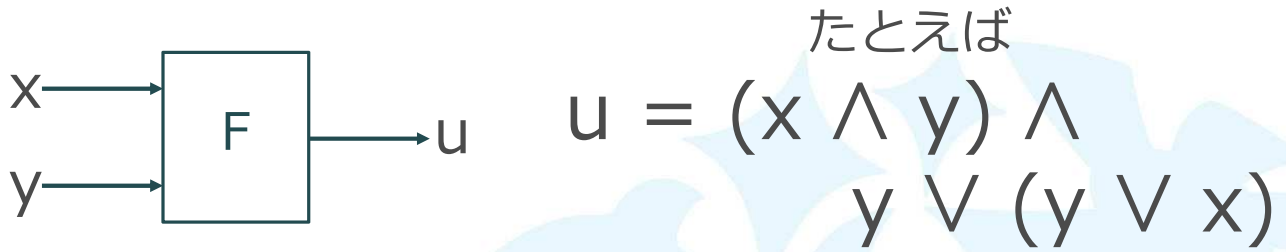
どんな組合せでもよい



東邦大学

3

複雑な論理演算



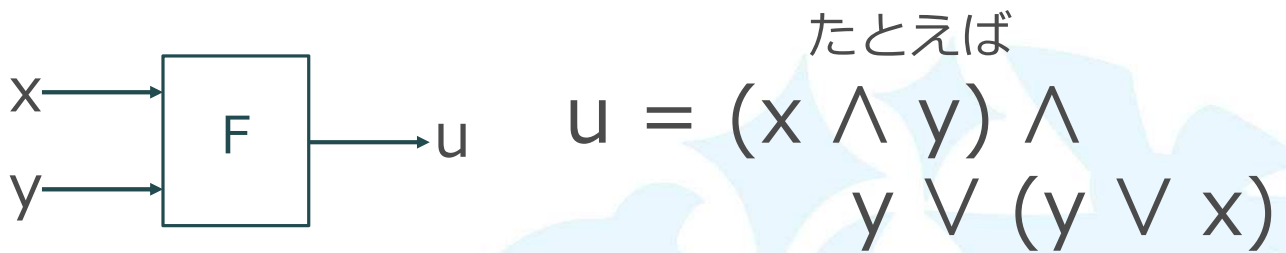
$F(x, y)$

$y \backslash x$	0	1
0	□	□
1	□	□

} どんな組合せでもよい



複雑な論理演算



$F(x, y)$

$y \backslash x$	0	1
0	□	□
1	□	□

どんな組合せでもよい

↑ どんな組合せでもよい

↑
 どういう
 関係か？

←
 2つは同じことか？
 どうすれば行き来できるか？



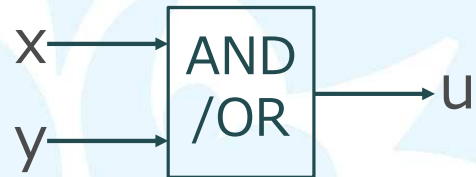
もっと複雑な論理関数とは

複雑な論理演算

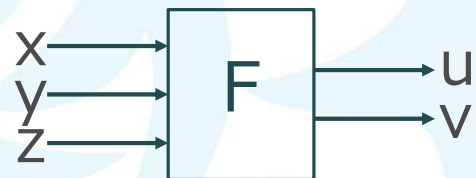
入出力が多い

たとえば

$$u = (x \wedge y) \wedge y \vee (y \vee x)$$



3入力2出力だと



東邦大学

6

3入力・1出力の関数

$$F(X, Y, Z)$$

X \ Y	0	1
0	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>



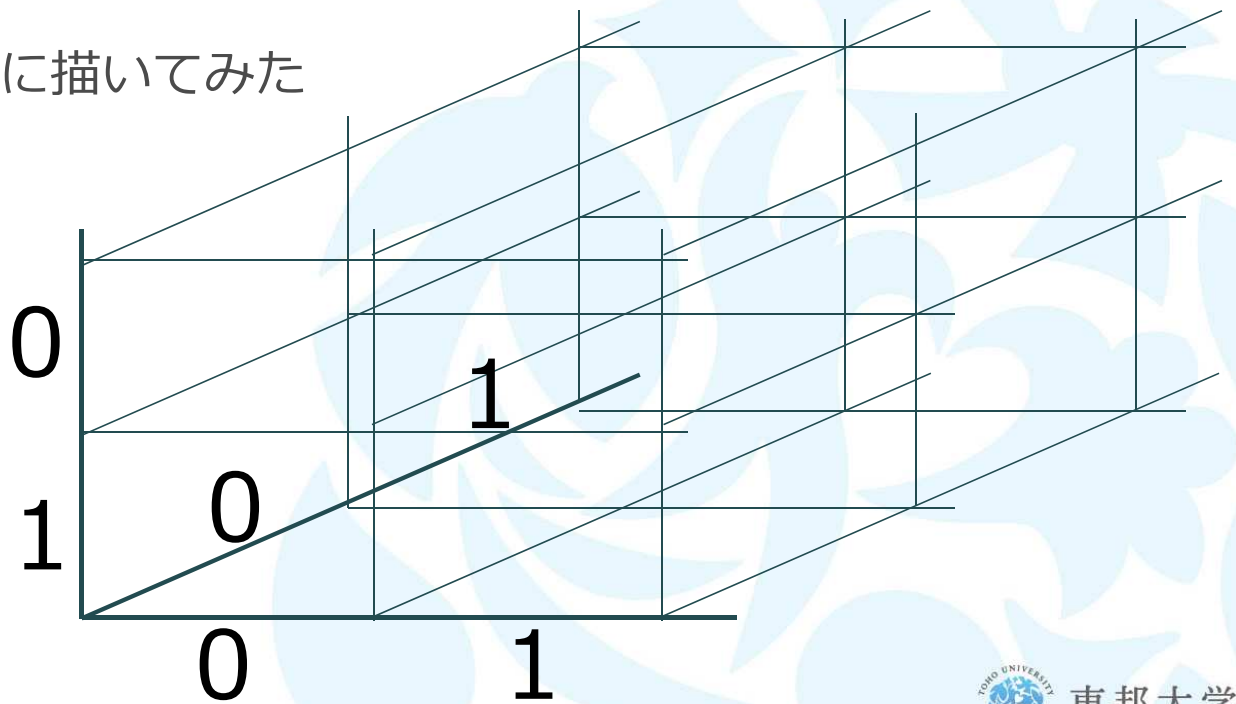
東邦大学

7

3入力・1出力の関数

$$F(X, Y, Z)$$

3次元に描いてみた



3入力・1出力の関数

$$F(X, Y, Z)$$

X	0		1	
Y \ Z	0	1	0	1
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3入力・1出力の関数

$$F(X, Y, Z) = (X \wedge Y) \vee Z$$

X	0		1	
Z \ Y	0	1	0	1
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



3入力・1出力の関数

$$F(X, Y, Z) = (X \wedge Y) \vee Z$$

X	0		1	
Z \ Y	0	1	0	1
0	0	0	0	1
1	1	1	1	1



3入力・1出力の関数

$$G(X, Y, Z) = (X \vee Y) \wedge Z$$

X	0		1	
Z \ Y	0	1	0	1
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



3入力・1出力の関数

$$G(X, Y, Z) = (X \vee Y) \wedge Z$$

X	0		1	
Z \ Y	0	1	0	1
0	0	0	0	0
1	0	1	1	1



(脱線) もう1つの真理値表の書き方

$$F(X,Y,Z) = (X \wedge Y) \vee Z$$

X	0		1	
Y	0	1	0	1
Z	0	1	0	1
0	0	0	0	1
1	1	1	1	1

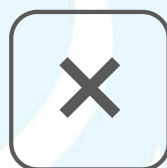
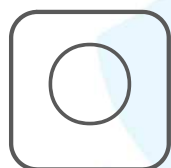
入力			出力
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



東邦大学

14

$\wedge \cdot \vee \cdot \neg$ を組合わせて
もっと複雑な関数を作れる
ことがわかりましたか？



↓
次へ



東邦大学

15

確認テストです

$H(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z)$
の真理値表を書いてください

16



東邦大学

解答

$(X \wedge Y) \vee (X \wedge Z)$

X	0		1	
Y	0	1	0	1
Z	0	0	0	1
0	0	0	0	1
1	0	0	1	1

入力			出力
x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

17



東邦大学

任意の関数の AND/OR/NOTによる合成

任意の論理関数

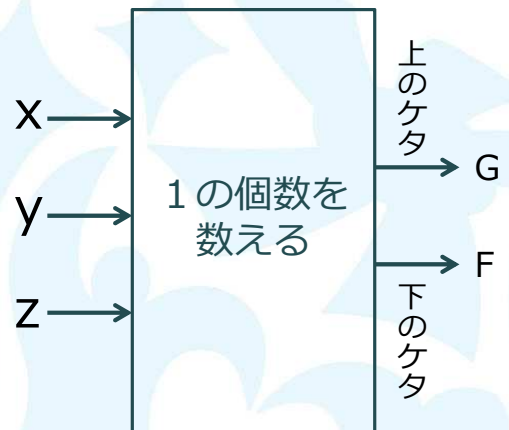
入力			出力
x	y	z	F
0	0	0	<input type="checkbox"/>
0	0	1	<input type="checkbox"/>
0	1	0	<input type="checkbox"/>
0	1	1	<input type="checkbox"/>
1	0	0	<input type="checkbox"/>
1	0	1	<input type="checkbox"/>
1	1	0	<input type="checkbox"/>
1	1	1	<input type="checkbox"/>

真理値表で
出力に任意の
値を書き込む

任意に作った論理関数の例

入力			出力	
x	y	z	G	F
0	0	0	<input type="checkbox"/>	<input type="checkbox"/>
0	0	1	<input type="checkbox"/>	<input type="checkbox"/>
0	1	0	<input type="checkbox"/>	<input type="checkbox"/>
0	1	1	<input type="checkbox"/>	<input type="checkbox"/>
1	0	0	<input type="checkbox"/>	<input type="checkbox"/>
1	0	1	<input type="checkbox"/>	<input type="checkbox"/>
1	1	0	<input type="checkbox"/>	<input type="checkbox"/>
1	1	1	<input type="checkbox"/>	<input type="checkbox"/>

←0個
←1個
←2個
←1個
←2個
←3個

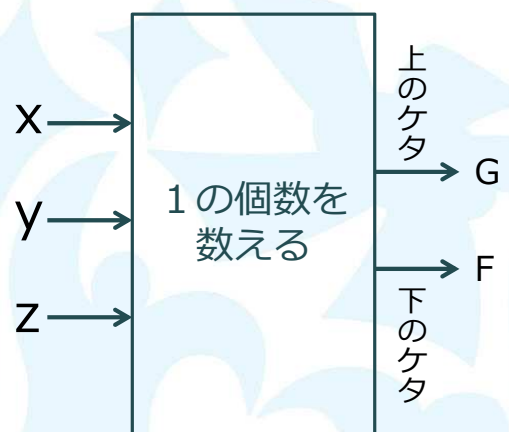


個数	G	F
0	0	0
1	0	1
2	1	0
3	1	1

任意に作った論理関数の例

入力			出力	
x	y	z	G	F
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

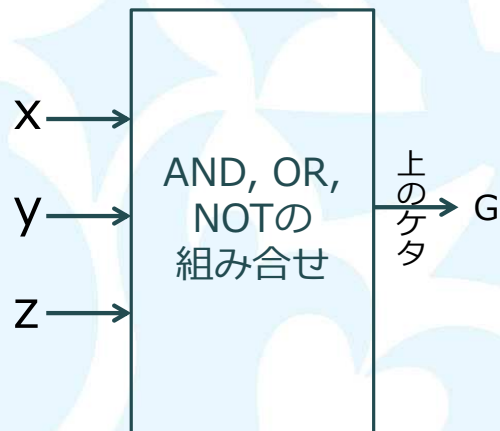
←0個
←1個
←2個
←1個
←2個
←3個



個数	G	F
0	0	0
1	0	1
2	1	0
3	1	1

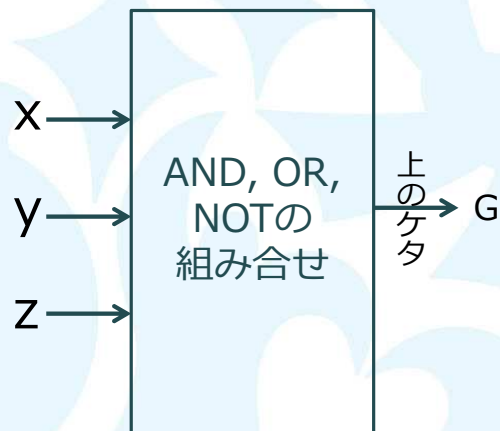
任意の論理関数を \wedge , \vee , \neg で合成

入力			出力
x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



任意の論理関数を \wedge , \vee , \neg で合成

入力			出力
x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



任意の論理関数を \wedge , \vee , \neg で合成

入力			出力
x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. 出力が1の行をマークする
2. マークした行で、それぞれの入力のパターンの時だけ1になる関数を作る

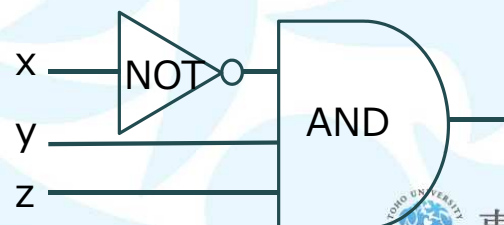
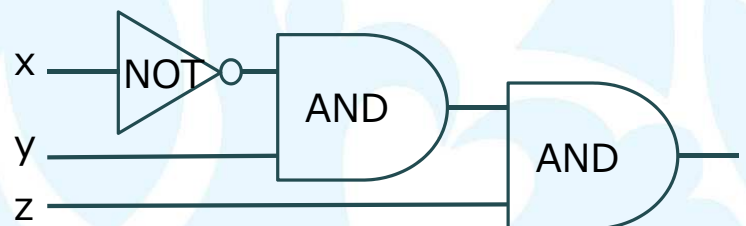


任意の論理関数を \wedge , \vee , \neg で合成

入力が 0, 1, 1 の時だけ出力が 1 になる関数を作る

入力			出力
x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

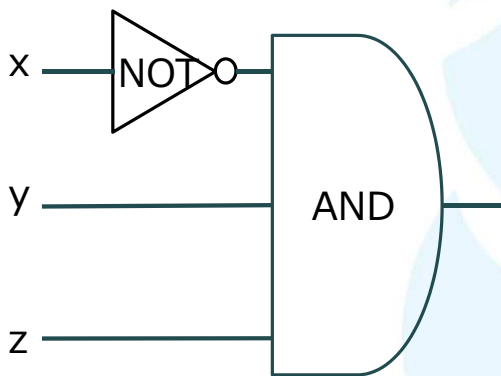
$$((\neg X) \wedge Y) \wedge Z$$



任意の論理関数を \wedge, \vee, \neg で合成

入力が 0, 1, 1 の時だけ出力が 1 になる関数を作る

$$(\neg X) \wedge Y \wedge Z$$



1. 0 である入力Xに対して NOTをかます
2. そうでない入力Y, Zには 何もしない
3. その結果の3本を、3入力のANDに入れる。式で書くと3つのANDを取る。

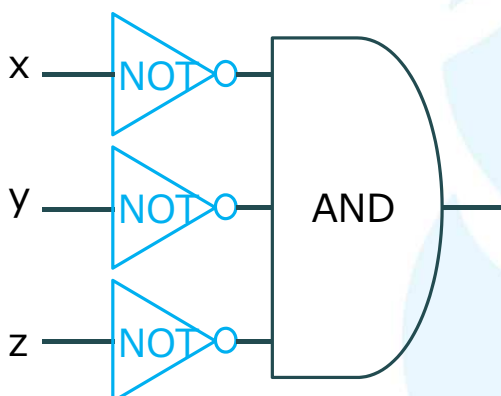


東邦大学

任意の論理関数を \wedge, \vee, \neg で合成

一般に、入力が $\bigcirc, \times, \triangle$ の時だけ出力が 1 になる関数を作る

$$([\neg]X) \wedge ([\neg]Y) \wedge ([\neg]Z)$$



1. 0 である入力に対して NOTをかます
2. そうでない入力には 何もしない
3. その結果の3本を、3入力のANDに入れる。

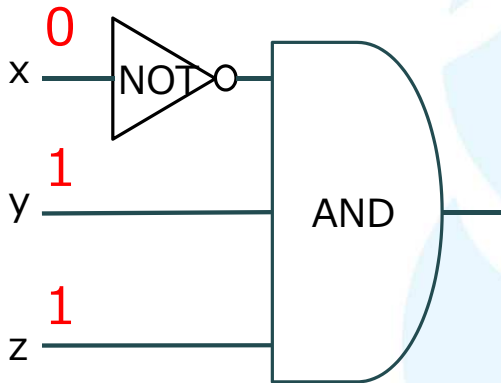


東邦大学

任意の論理関数を \wedge, \vee, \neg で合成

\bigcirc \times \triangle
 入力が **0, 1, 1** の時だけ出力が 1 になる関数を作る

\neg あり \neg なし \neg なし
 $([\neg]X) \wedge ([\neg]Y) \wedge ([\neg]Z)$



1. 0である入力に対して NOTをかます
2. そうでない入力には 何もしない
3. その結果の3本を、3入力のANDに入れる。

任意の論理関数を \wedge, \vee, \neg で合成

入力			出力
x	y	z	G
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

→ 入力が0, 1, 1の時だけ出力1

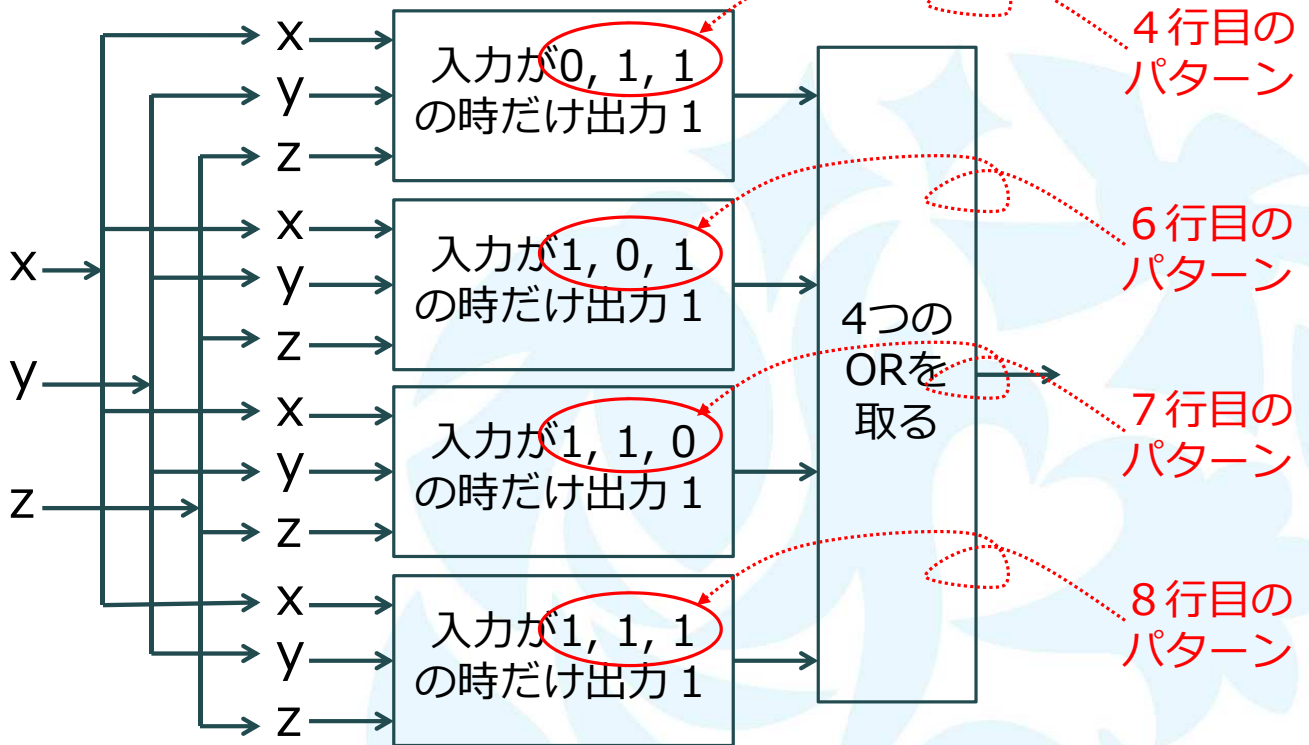
→ 入力が1, 0, 1の時だけ出力1

→ 入力が1, 1, 0の時だけ出力1

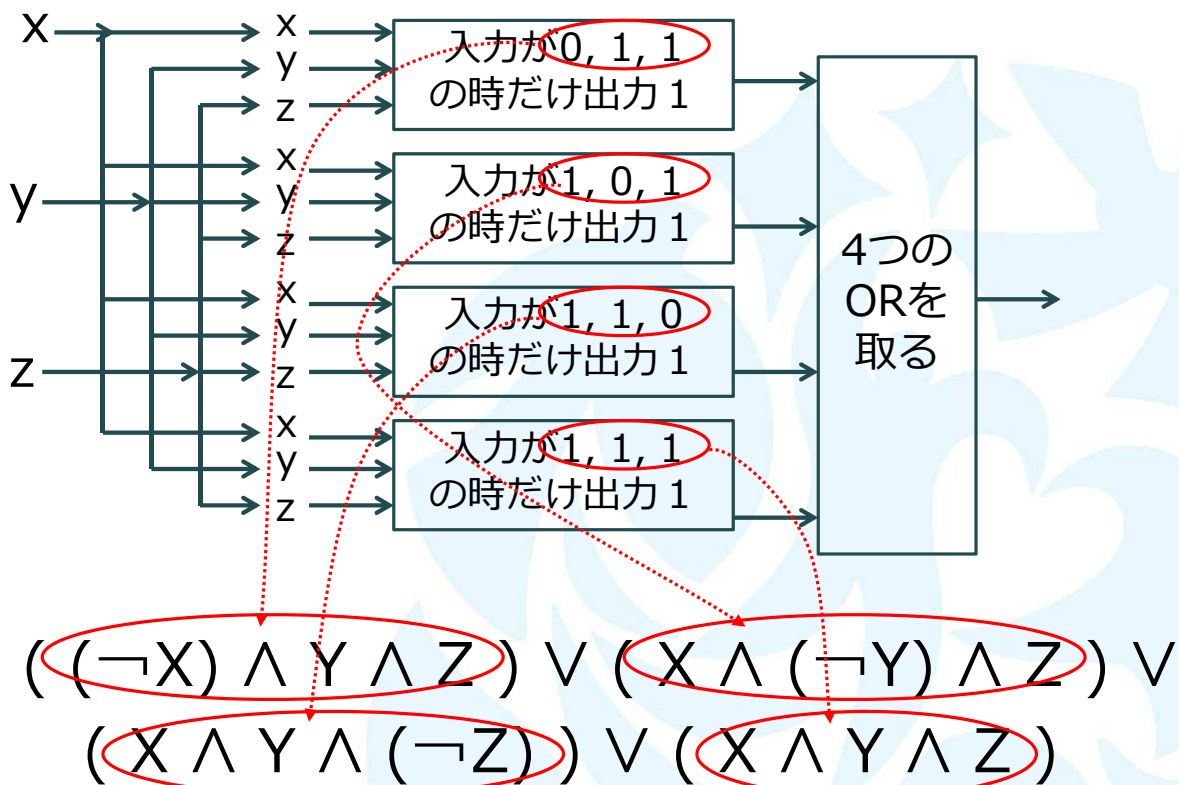
→ 入力が1, 1, 1の時だけ出力1

組み合わせる

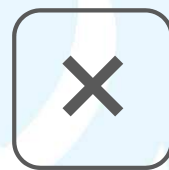
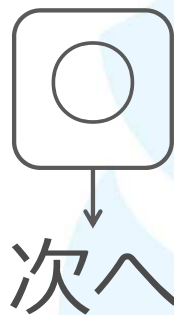
任意の論理関数を \wedge , \vee , \neg で合成



任意の論理関数を \wedge , \vee , \neg で合成



任意の論理関数が
 $\wedge \vee \neg$ 演算を組合わせて作れる
ことがわかりましたか？



確認テストです

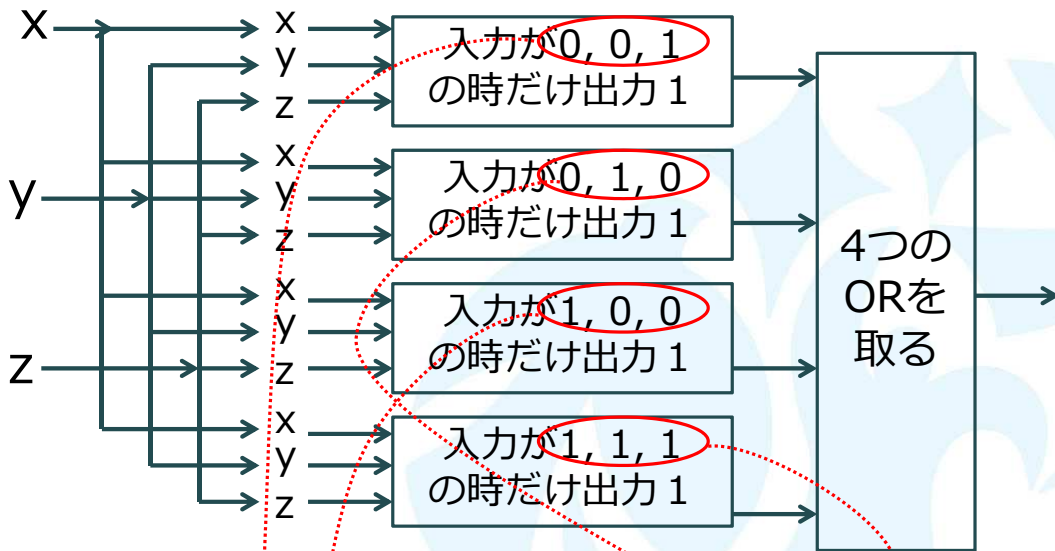
入力			出力
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

左の真理値表は
前に出た「1の個数」
を数える関数のうち
下(右)の桁Fです。

この論理関数を
 $\wedge \vee \neg$ の組合せで
書いてください。



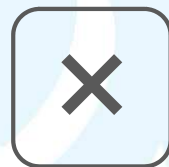
確認テスト



$$\left((\neg X) \wedge (\neg Y) \wedge Z \right) \vee \left((\neg X) \wedge Y \wedge (\neg Z) \right) \vee \left(X \wedge (\neg Y) \wedge (\neg Z) \right) \vee \left(X \wedge Y \wedge Z \right)$$



任意の論理関数を
 $\wedge \vee \neg$ 演算を組合わせて
作ることができますか？



次へ

