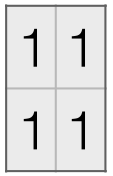
Informatics 1

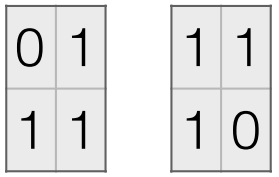
Computation and Logic

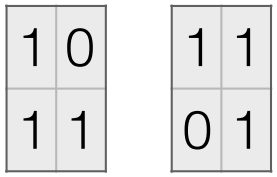
Entailment

Michael Fourman

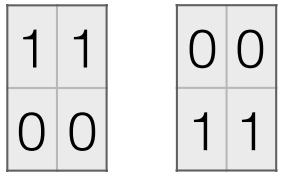
Exercise 1.2

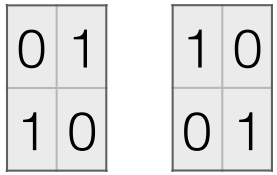


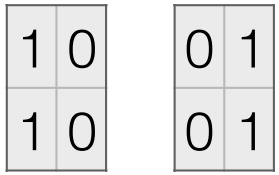
$A \vee B$ 



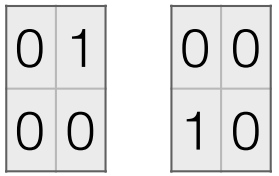
$A \to B$

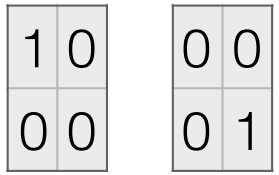
$\neg A$ 



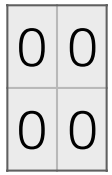


B

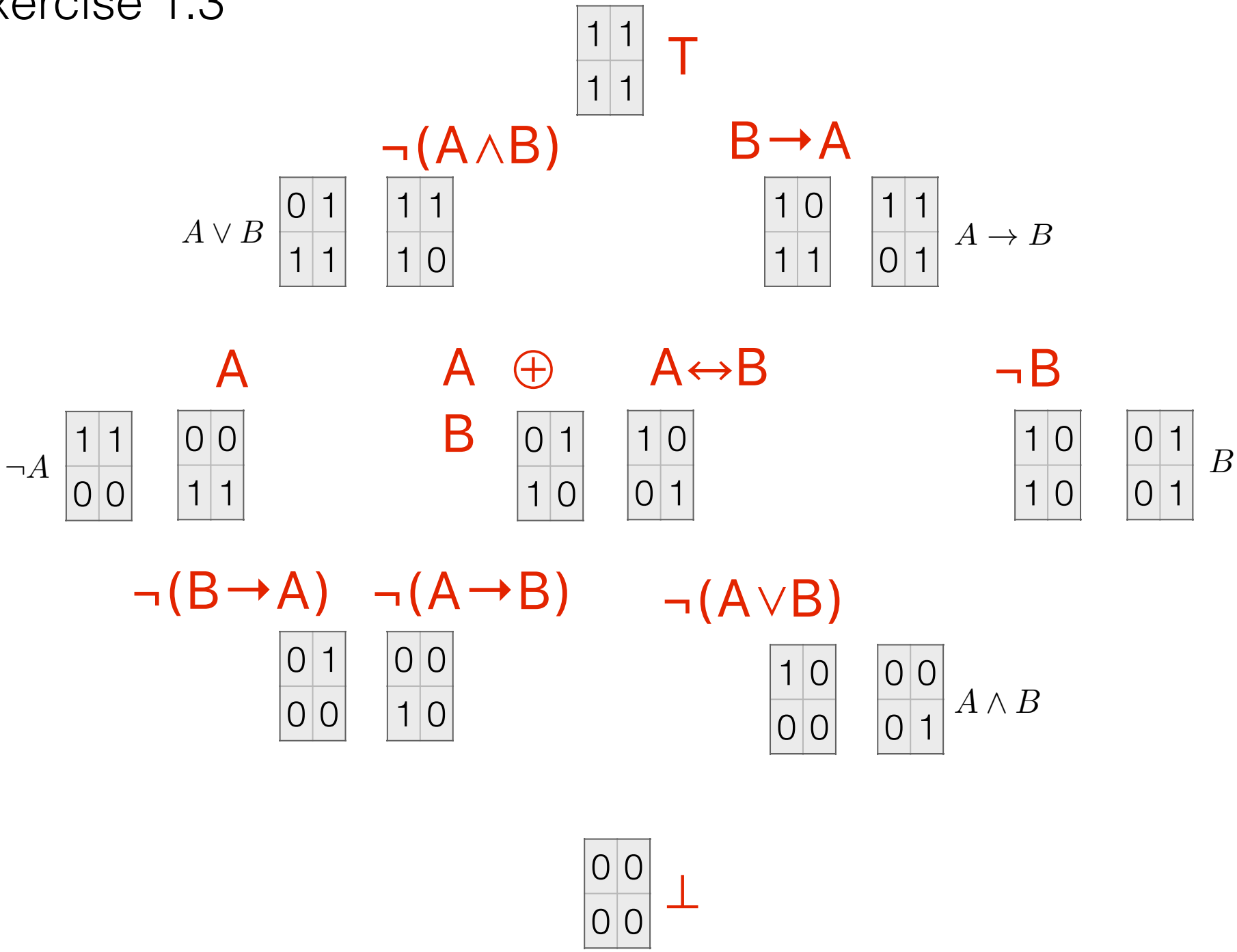


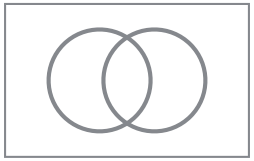


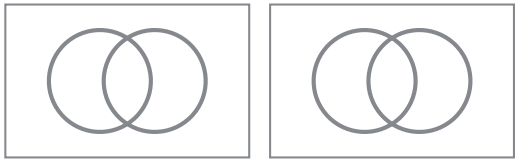
$A \wedge B$

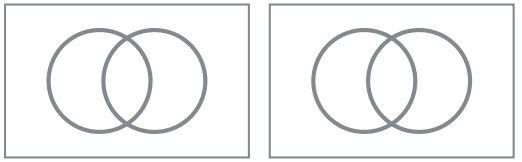


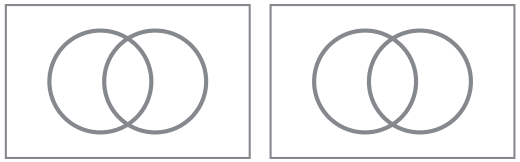
Exercise 1.3

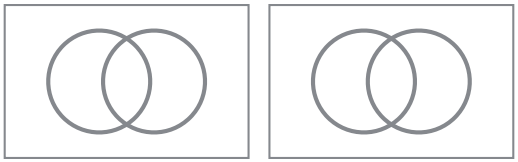

1 1
1 1
⊤
¬(A∧B)
B→A
A ∨ B
0 1
1 1
1 1
1 0
1 0
1 1
1 1
0 1
A → B
A
A ⊕
A↔B
¬B
¬A
1 1
0 0
0 0
1 1
A
B
0 1
1 0
1 0
0 1
1 0
1 0
0 1
0 1
B
¬(B→A)
¬(A→B)
¬(A∨B)
0 1
0 0
0 0
1 0
1 0
0 0
0 0
0 1
A ∧ B
0 0
0 0
⊥

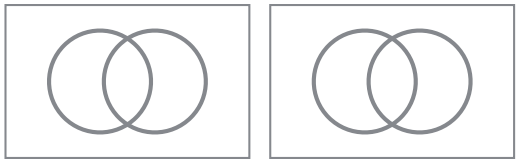


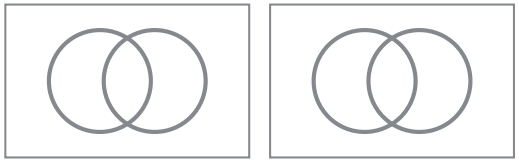


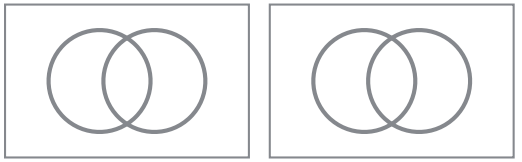


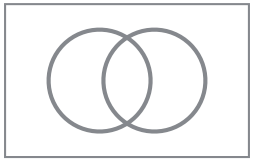


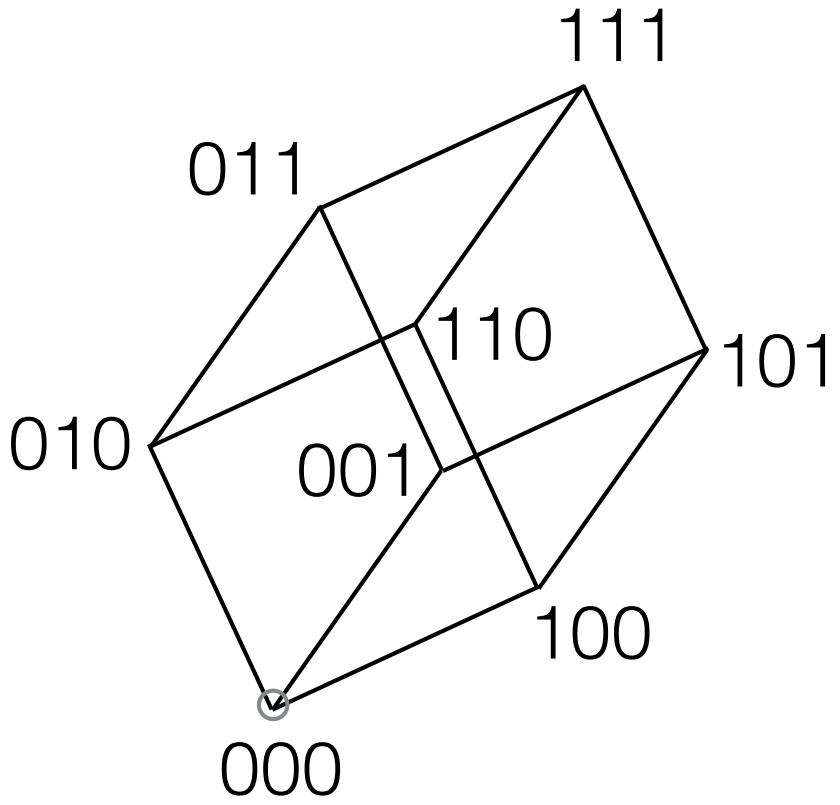




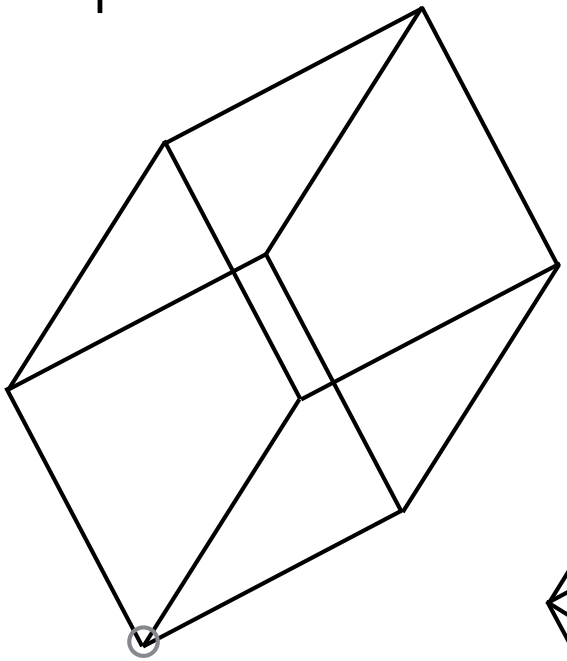




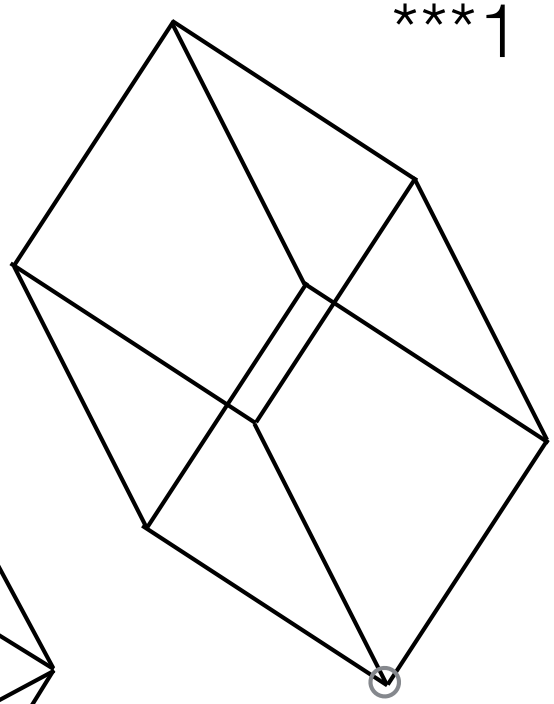


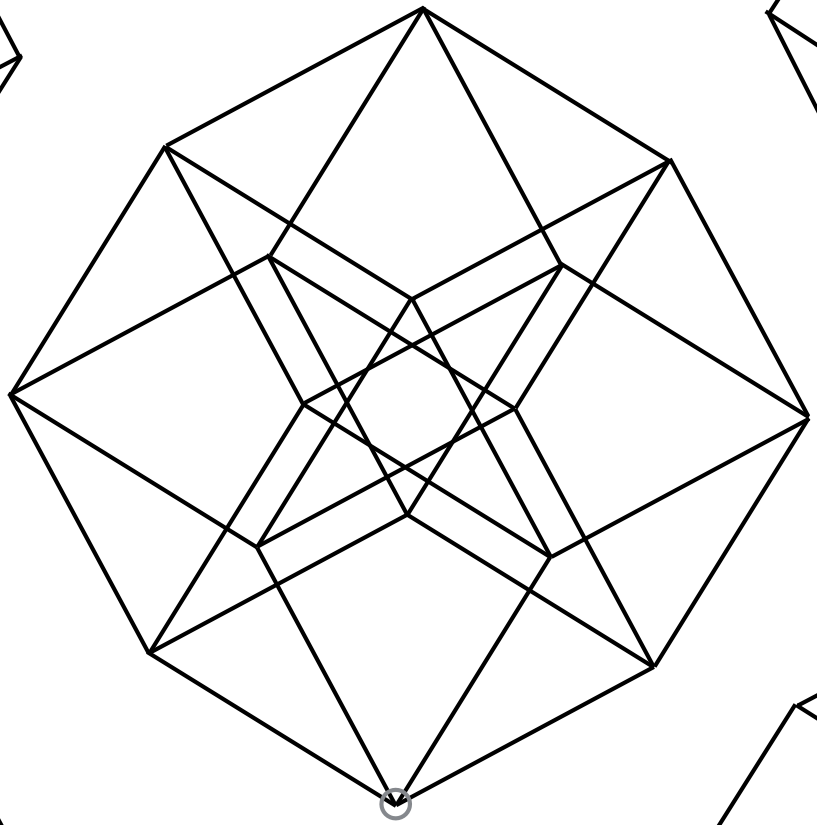

111
011
110
101
010
001
100
000

1***

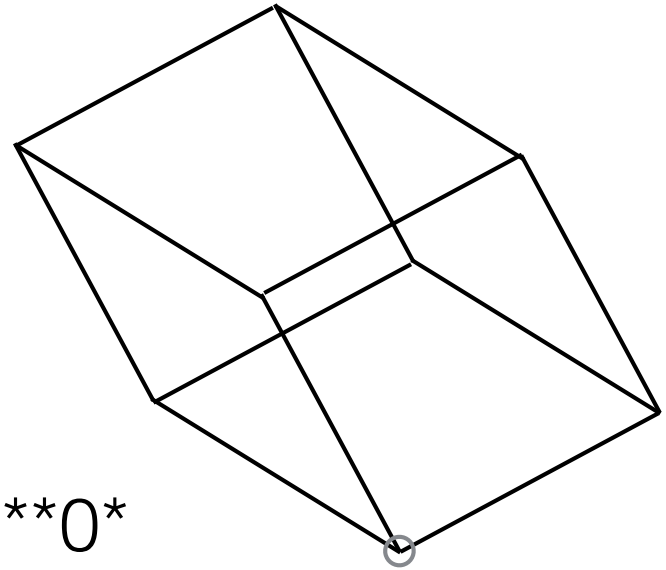


***1

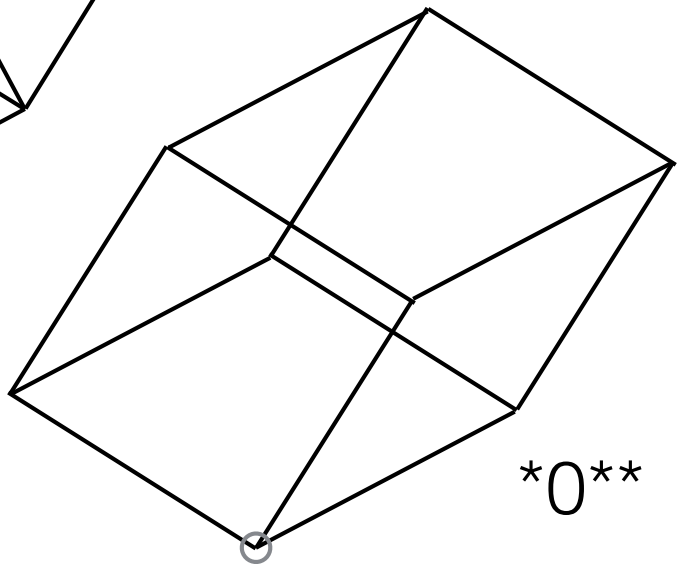




**0*

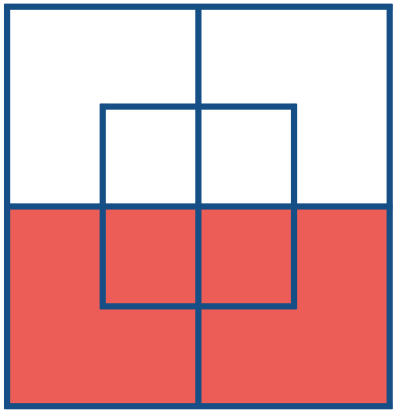


*0**

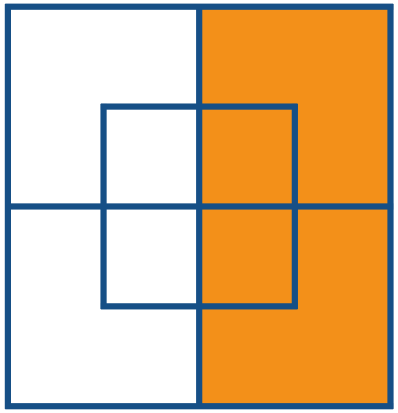


Lewis Carroll (The Rev. C.L. Dodgson)

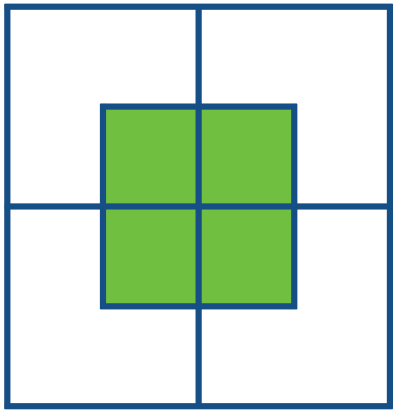
R



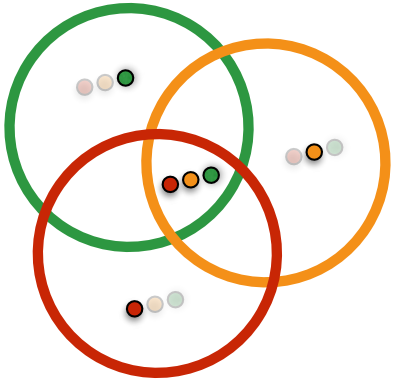
A

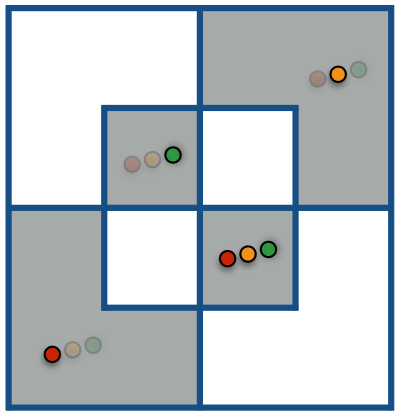


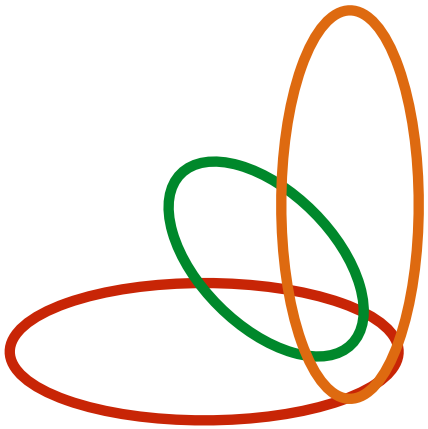
G

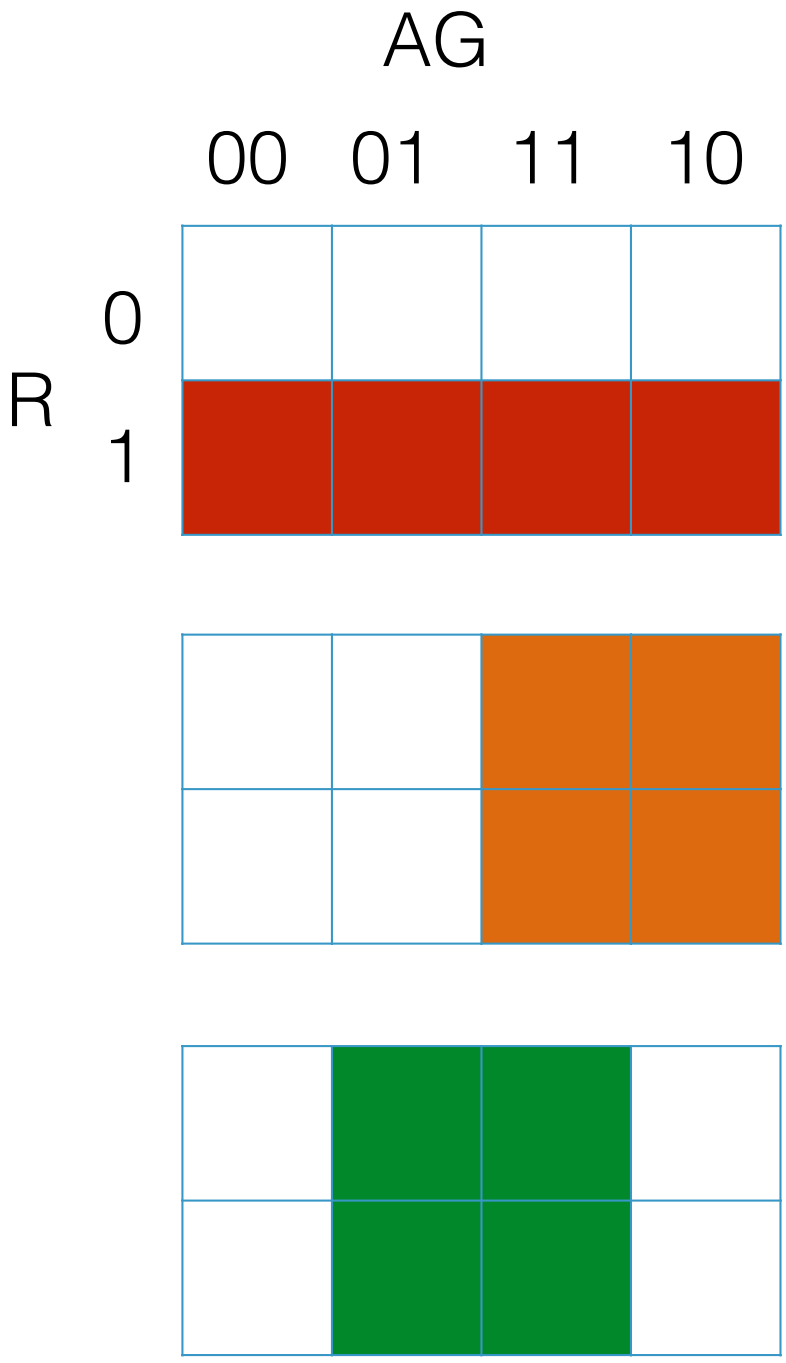


$$\{x \mid G(x) \leftrightarrow R(x) \leftrightarrow A(x)\}$$

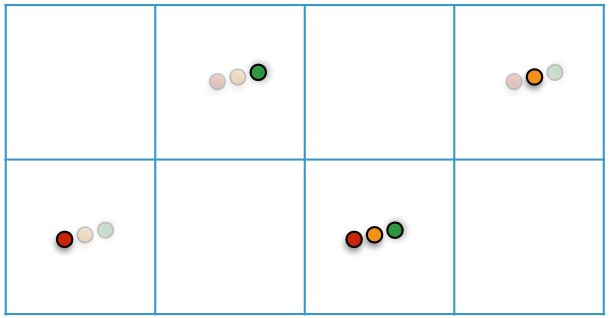


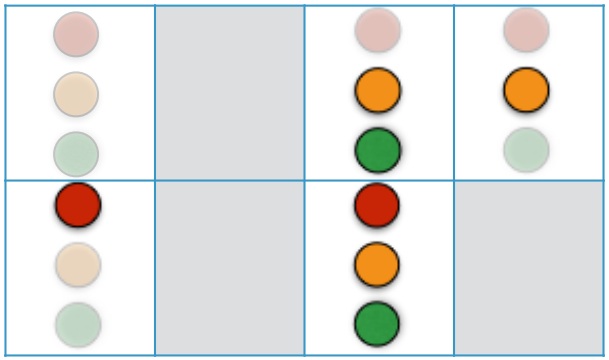


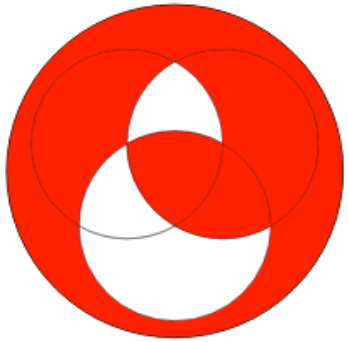




$$\{x \mid G(x) \leftrightarrow R(x) \leftrightarrow A(x)\}$$

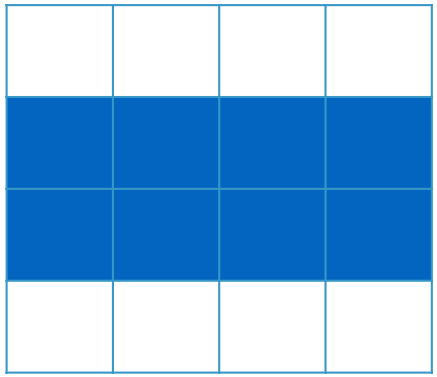


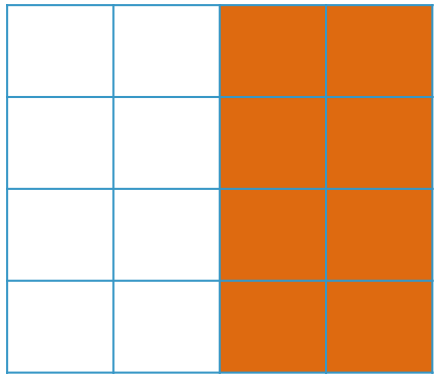


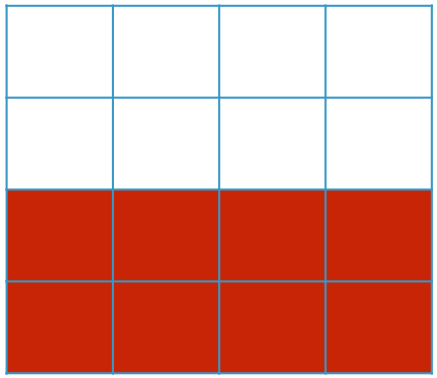


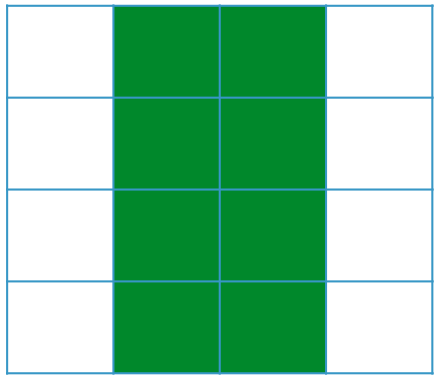
Karnaugh Maps

RB \ AG	00	01	11	10
00				
01				
11				
10				

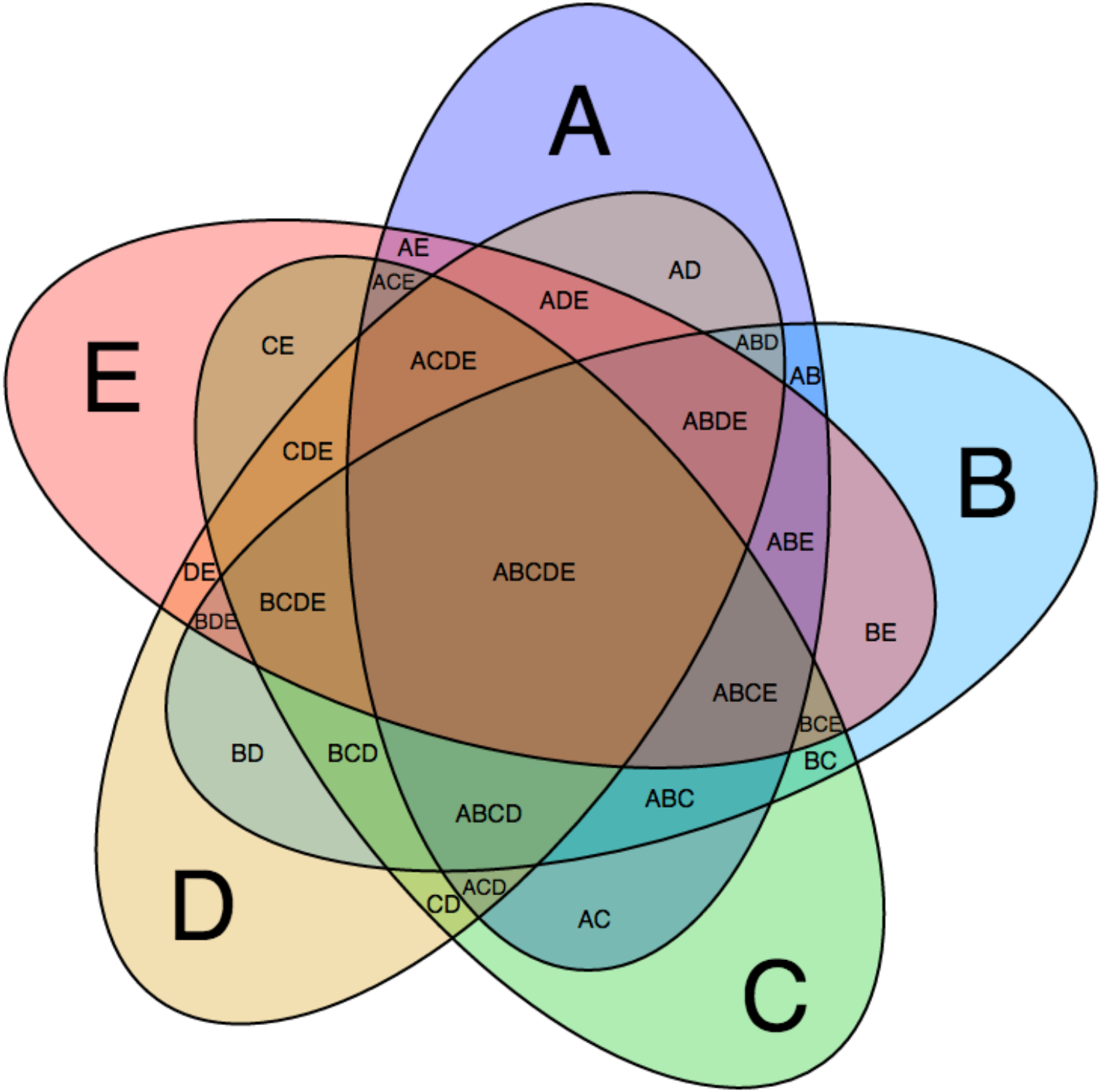


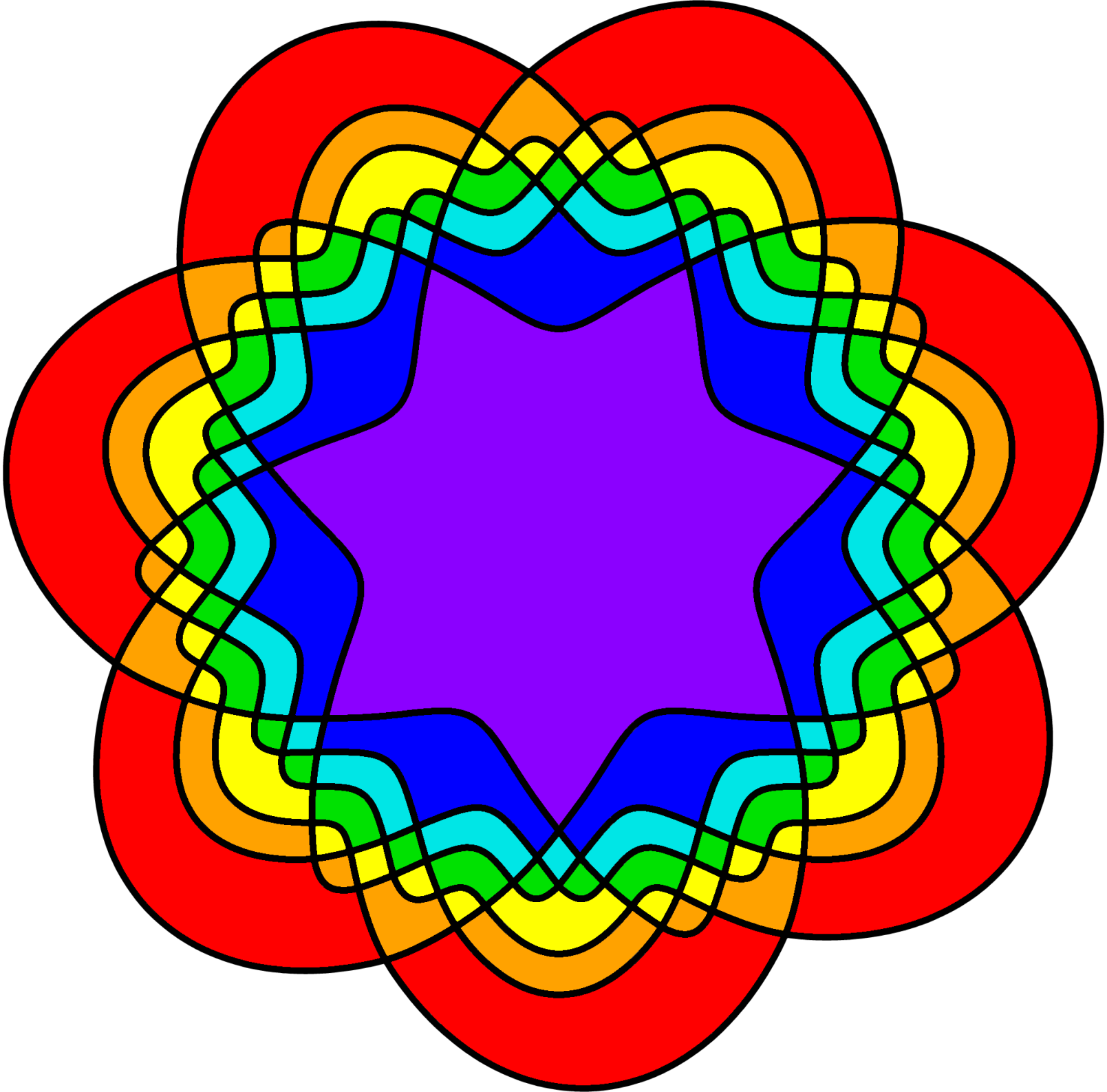






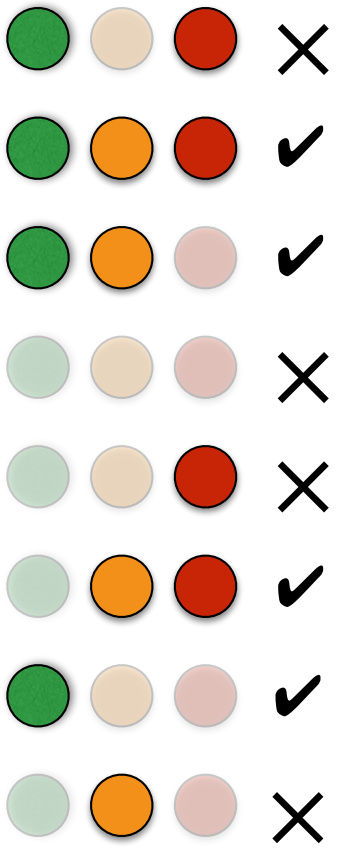
4 atoms:16 states: 64K subsets

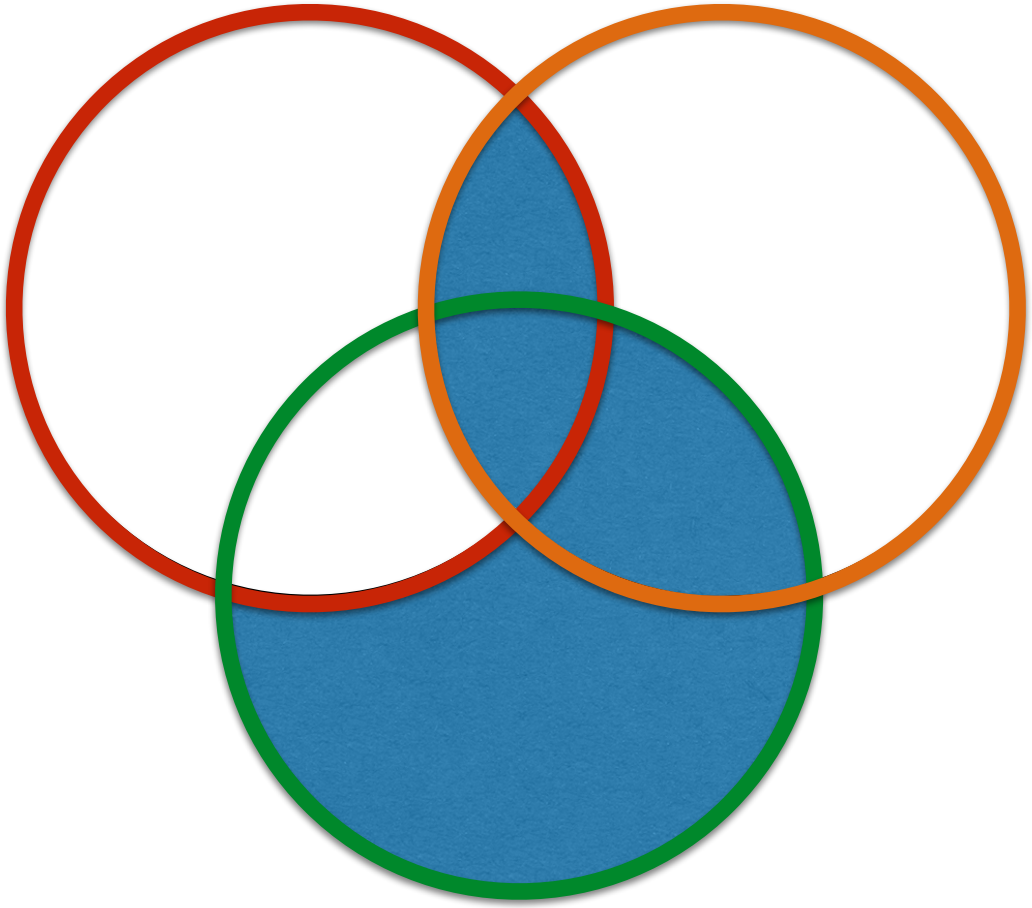

A
E
B
D
C
AE
ACE
AD
ADE
CE
ACDE
ABD
AB
ABDE
CDE
ABE
DE
ABCDE
BCDE
BDE
BE
ABCE
BCE
BD
BCD
BC
ABC
ABCD
ACD
CD
AC

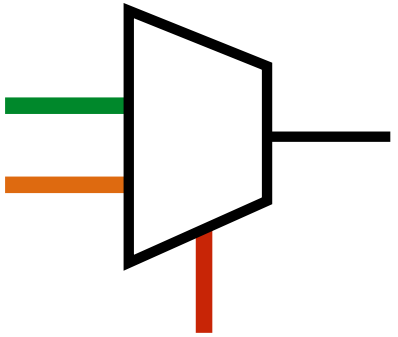


(R?A:G)

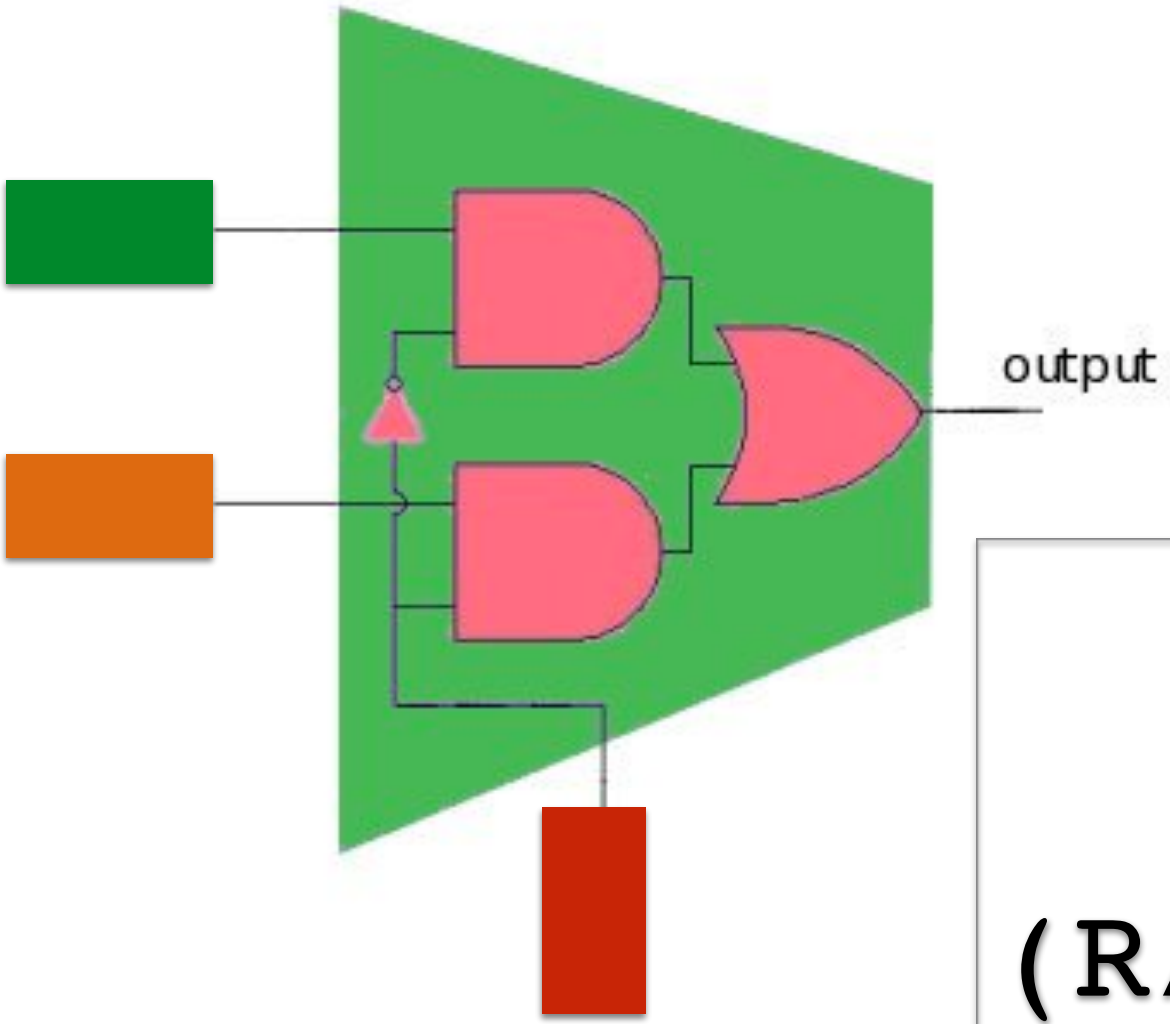
if R then A else G

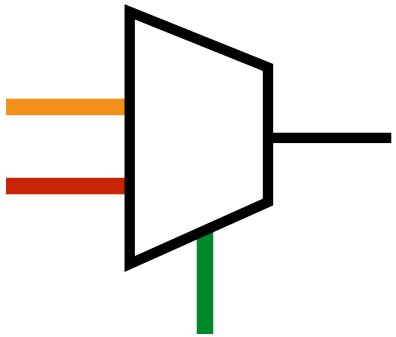






multiplexer – ITE

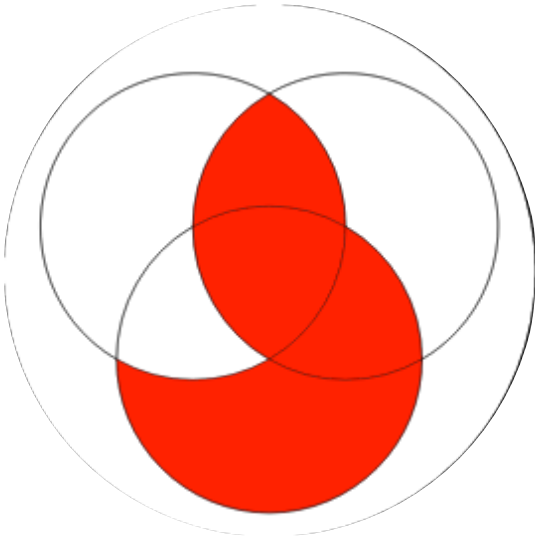




$$(R?A:G)$$

$$=$$

$$(R \wedge A) \vee (\neg R \wedge G)$$



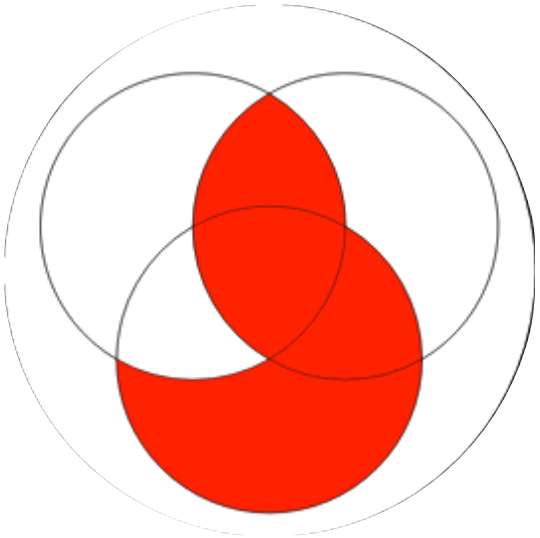
R?⊥:⊤

R?⊤:A

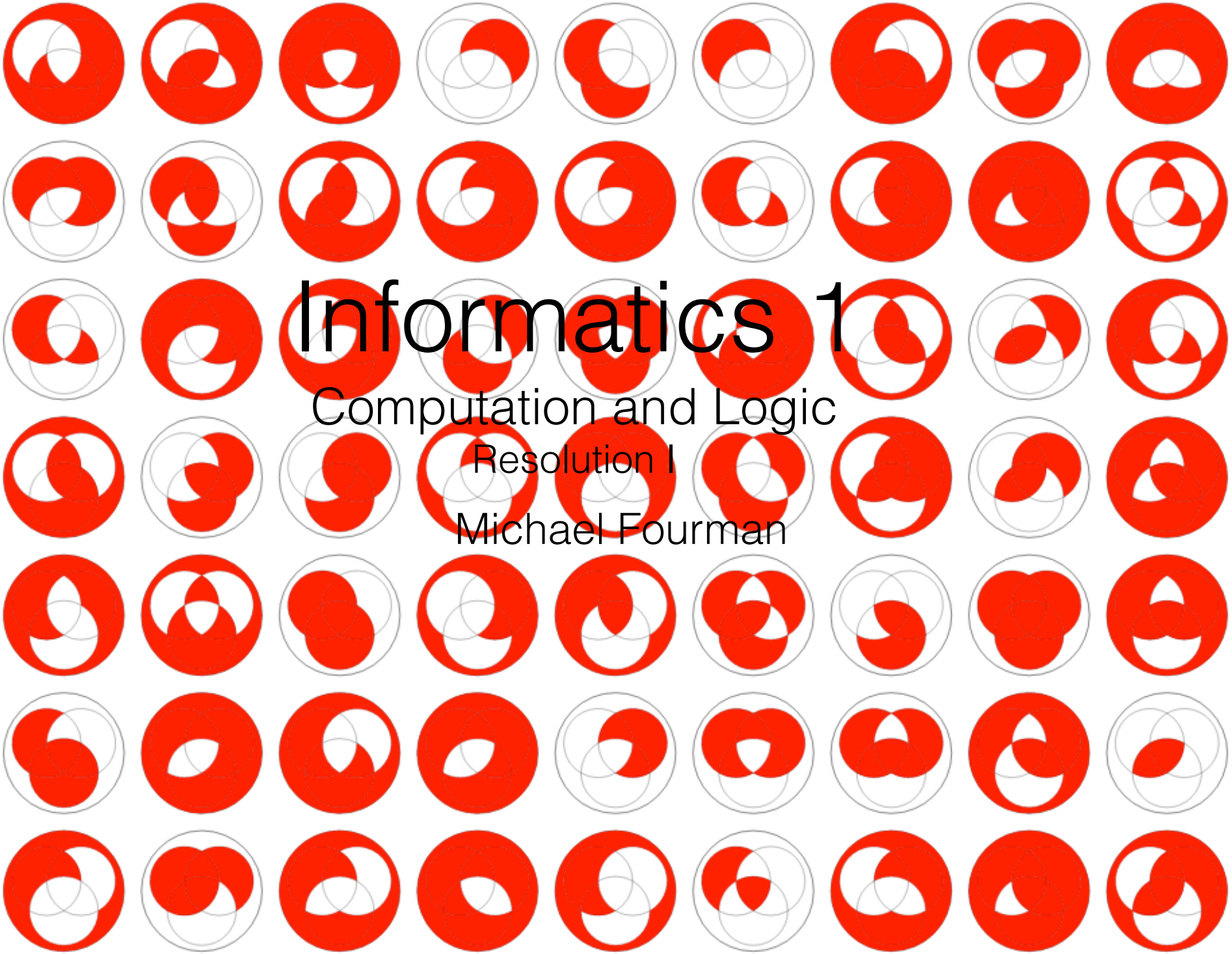
R?A:⊥

R?A:⊤

R?⊥:A



$R?\bot:\top$	$\neg R$
$R?\top:A$	$R \vee A$
$R?A:\bot$	$R \wedge A$
$R?A:\top$	$R \rightarrow A$
$R?\bot:A$	$\neg(A \rightarrow R)$

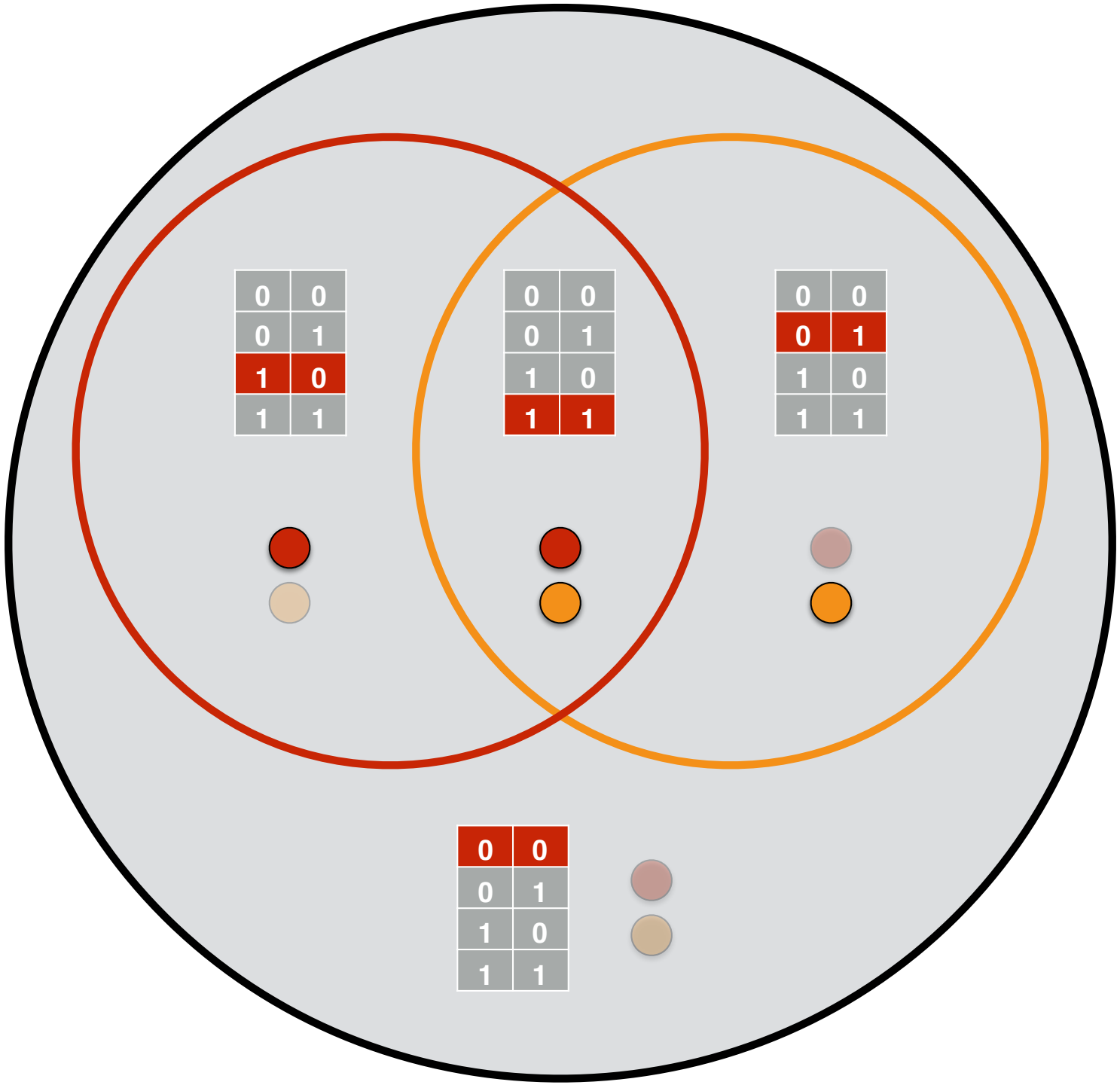


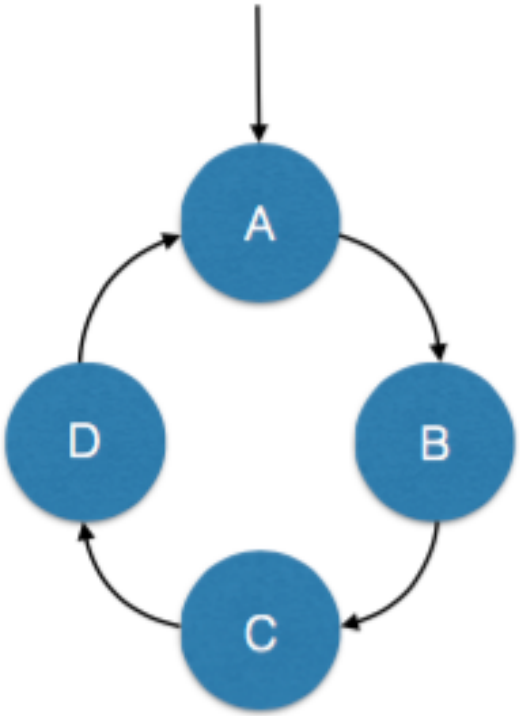
Informatics 1

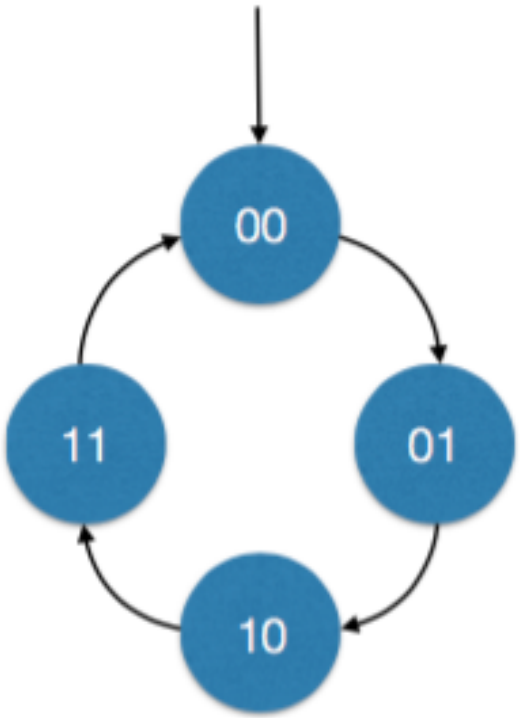
Computation and Logic

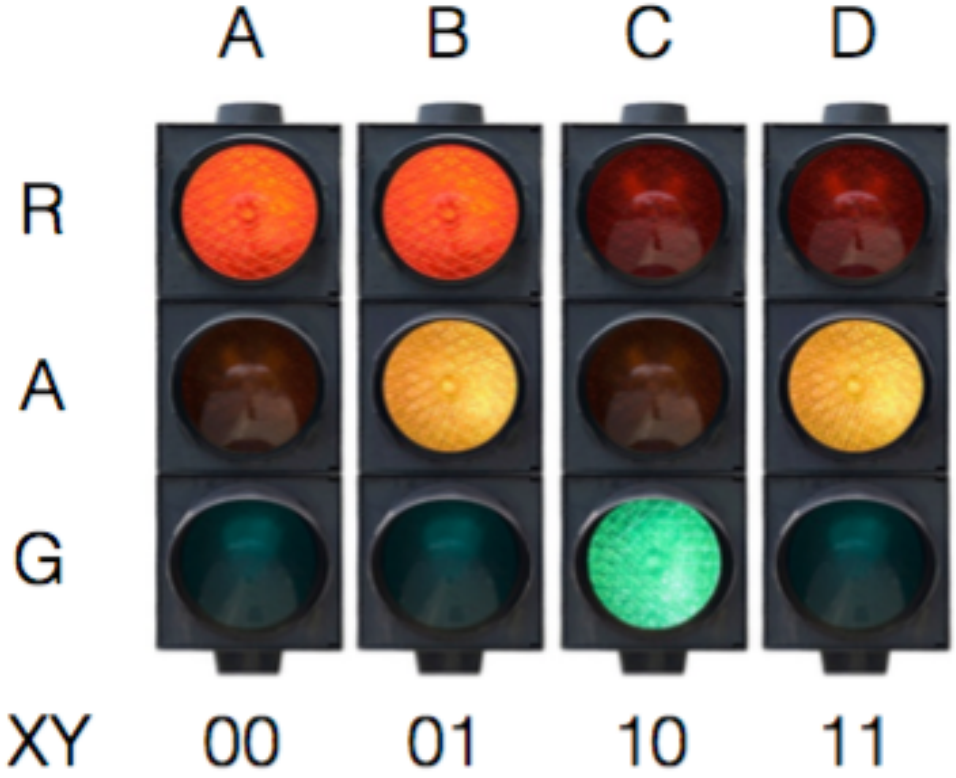
Resolution I

Michael Fourman


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







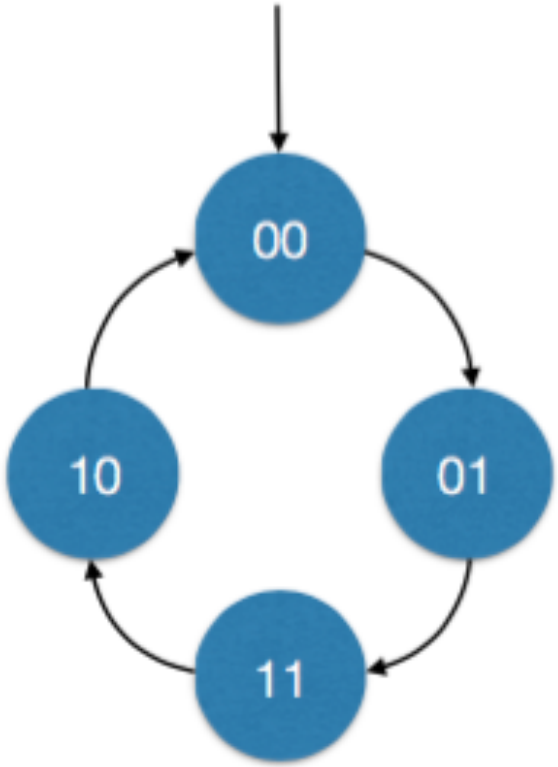
The traffic light has only four states.
the diagram shows a two-bit encoding of these four states. If we call the two bits X and Y then the next state logic can be given by

$$X' = X \oplus Y \text{ and } Y' = \neg Y$$

and the output logic (the signals to the lights) by

$$R = \neg X \quad A = Y \quad G = \neg X \wedge Y$$

This question concerns a different two-bit encoding of the four states, as shown below.

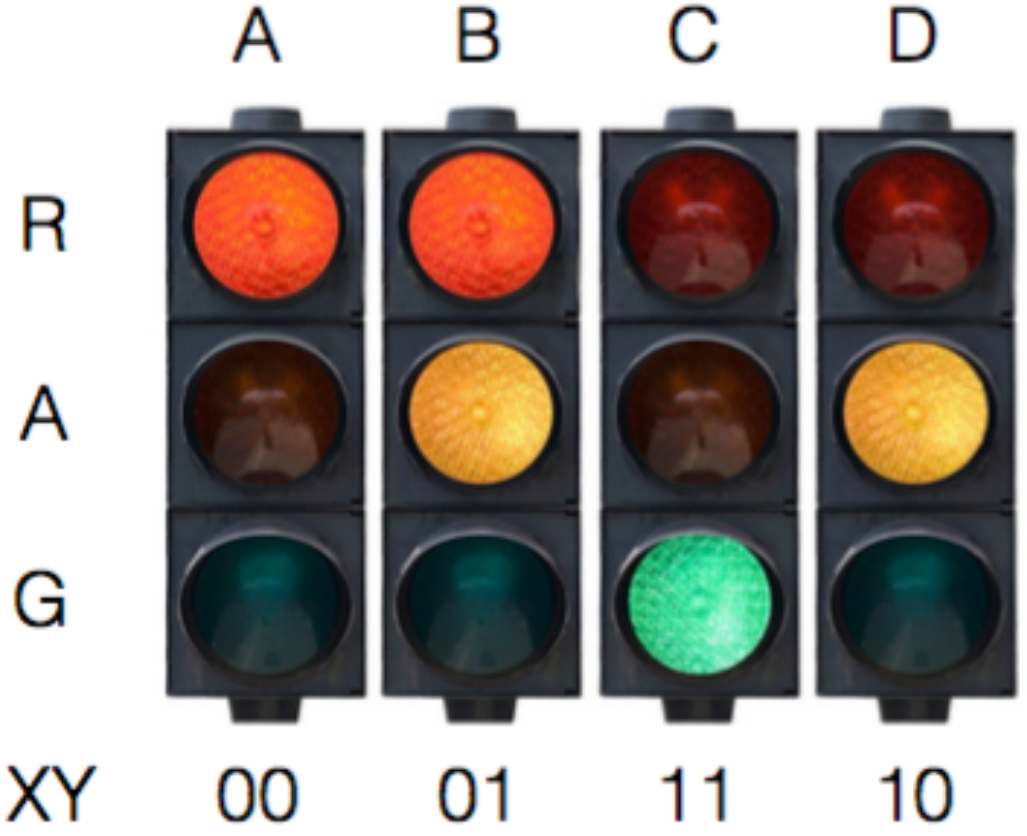


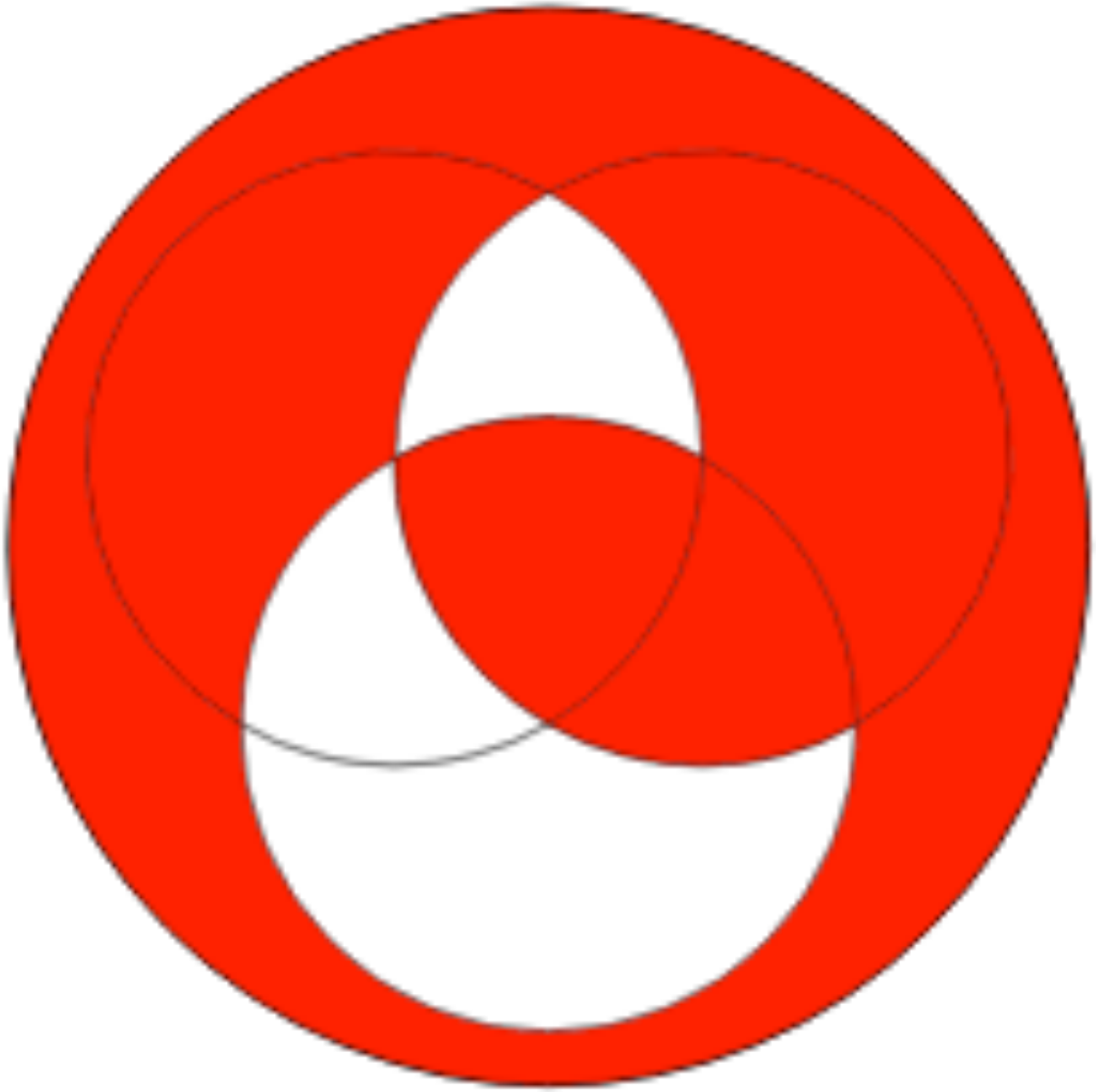
Give expressions for the next state logic

$X' =$ Y $\quad Y' =$ $\neg X$

and the output logic

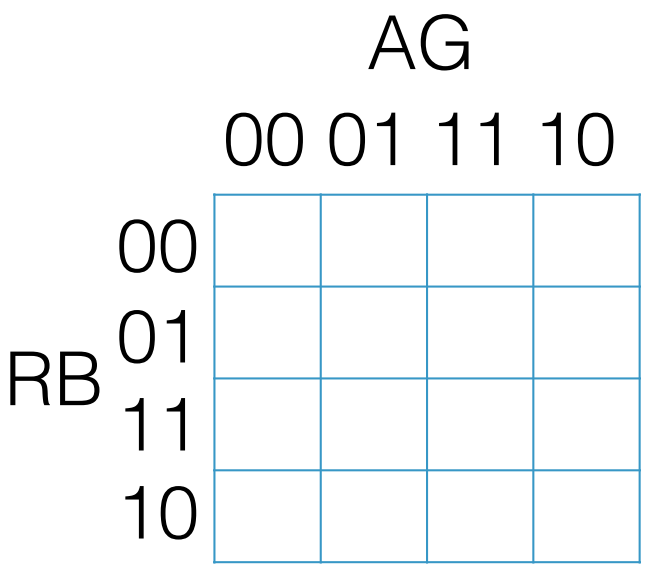
$R =$ $\neg X$ $\quad A =$ $X \oplus Y$ $\quad G =$ $X \wedge Y$

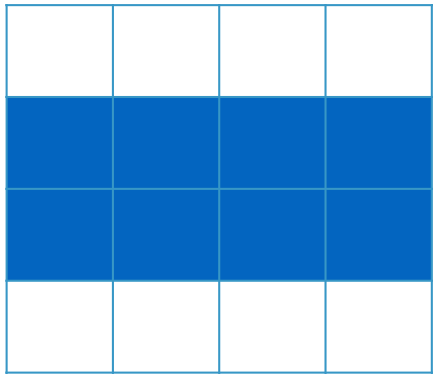


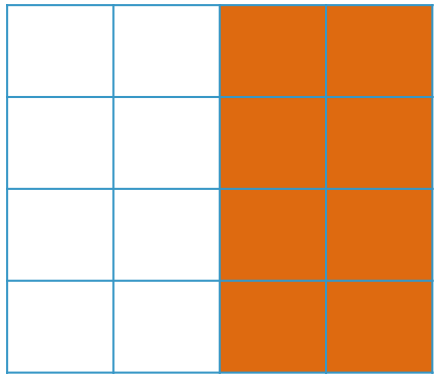


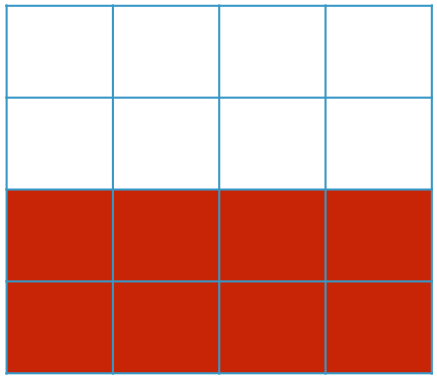
R	A	G	??
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

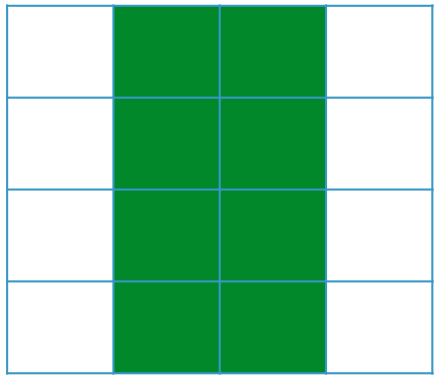
Karnaugh Maps











4 atoms:16 states: 64K subsets