Informatics 1

Computation and Logic

Sets of States: Venn Diagrams and Truth Tables

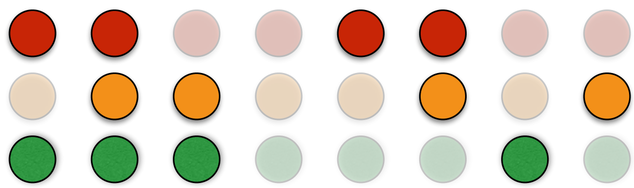
Michael Fourman
@mp4man

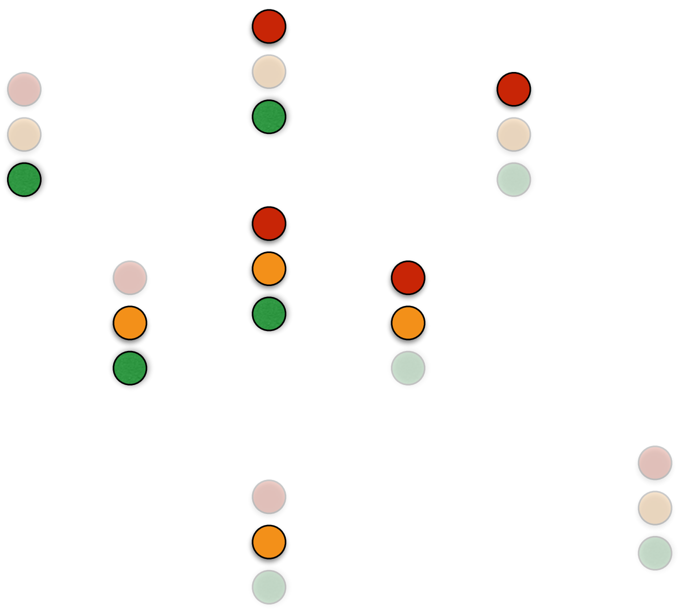
properties & sets

boolean circuits

boolean formulæ

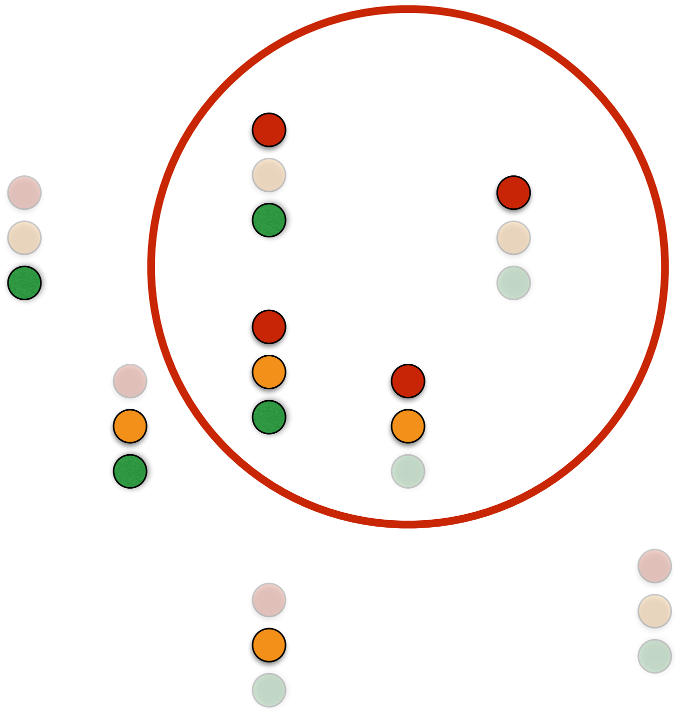
boolean functions





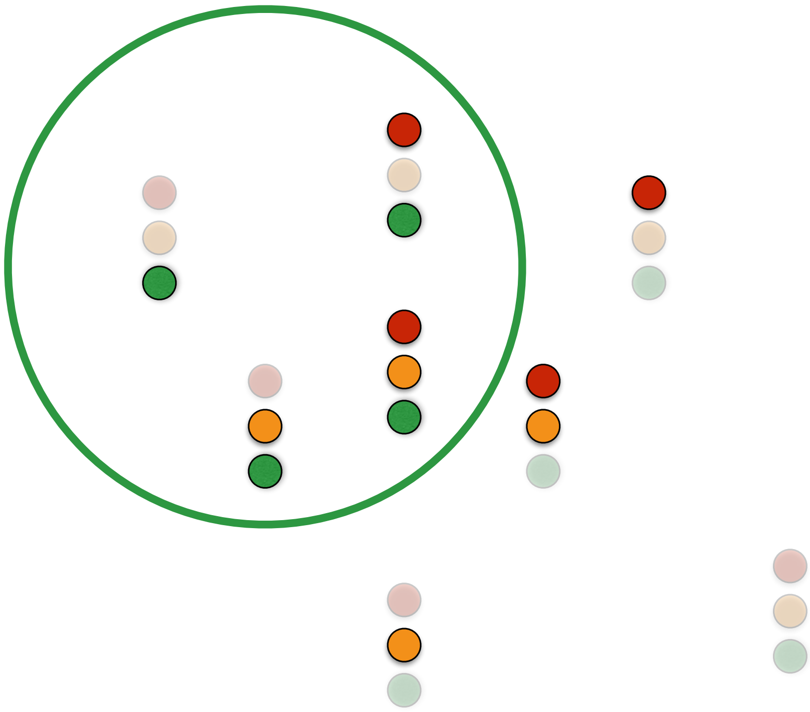
$$\{x \mid R(x)\}$$

R	A	G
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



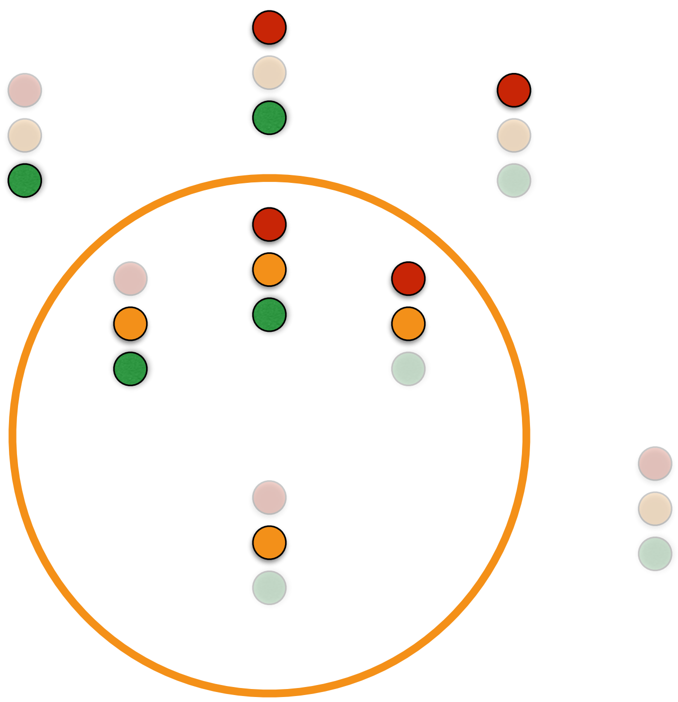
$$\{x \mid G(x)\}$$

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

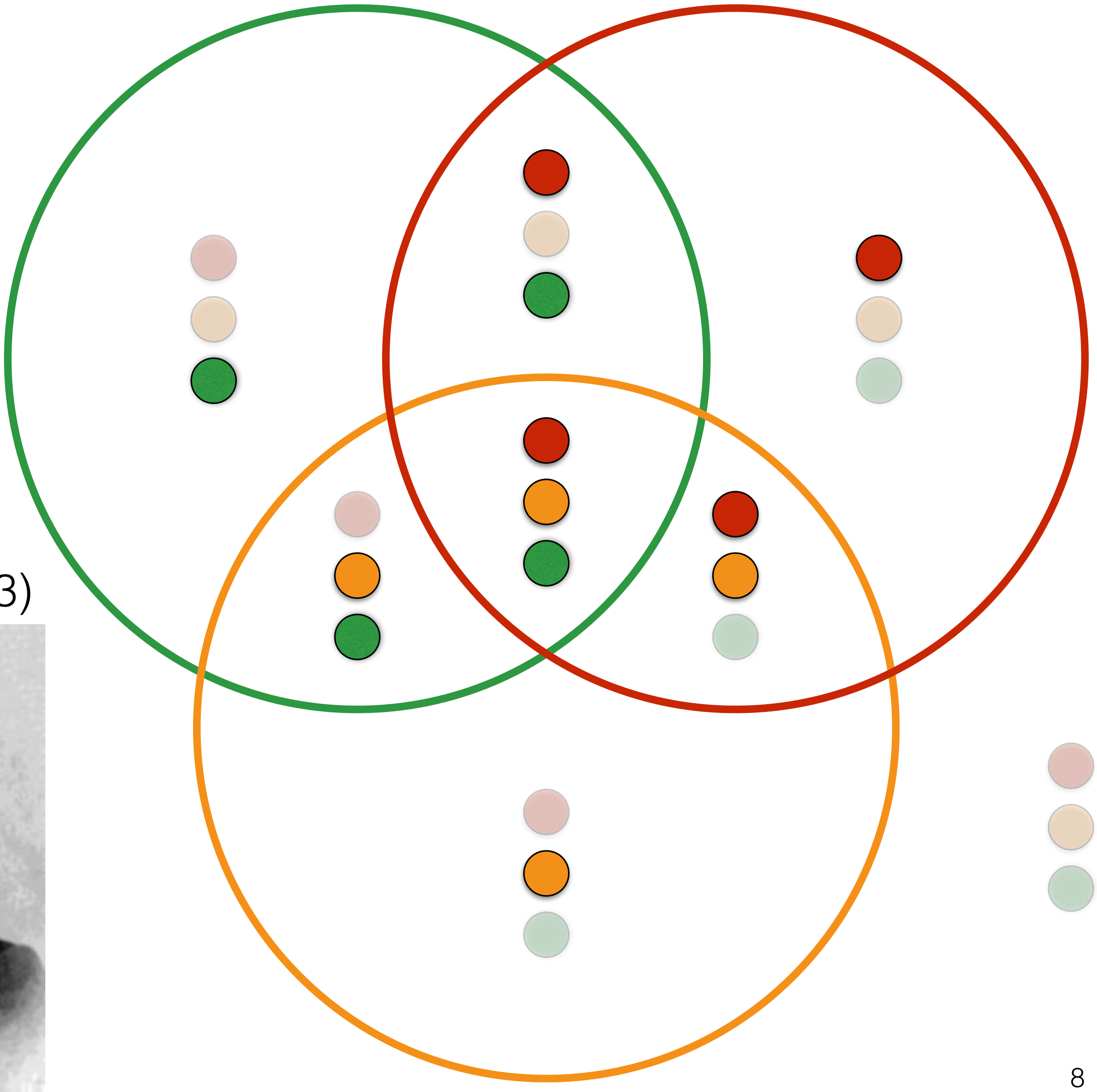


$$\{x \mid A(x)\}$$

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



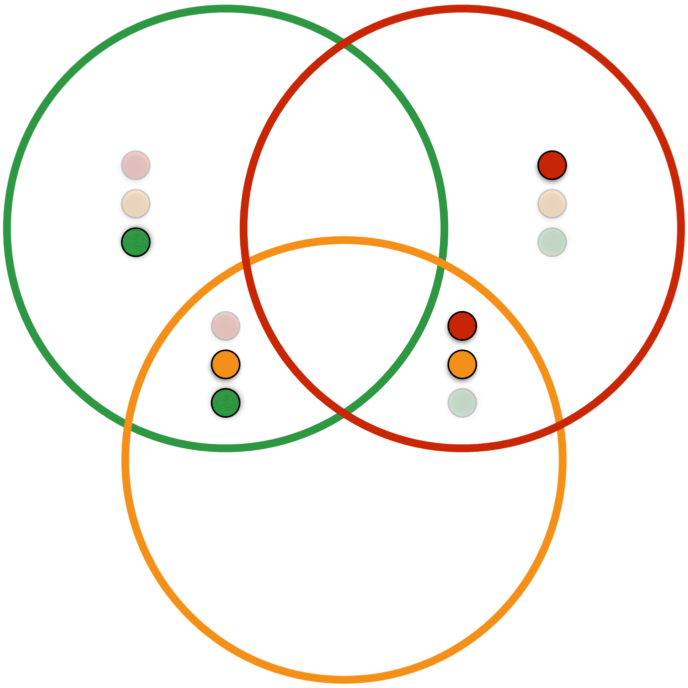
Venn (1834–1923)



xor

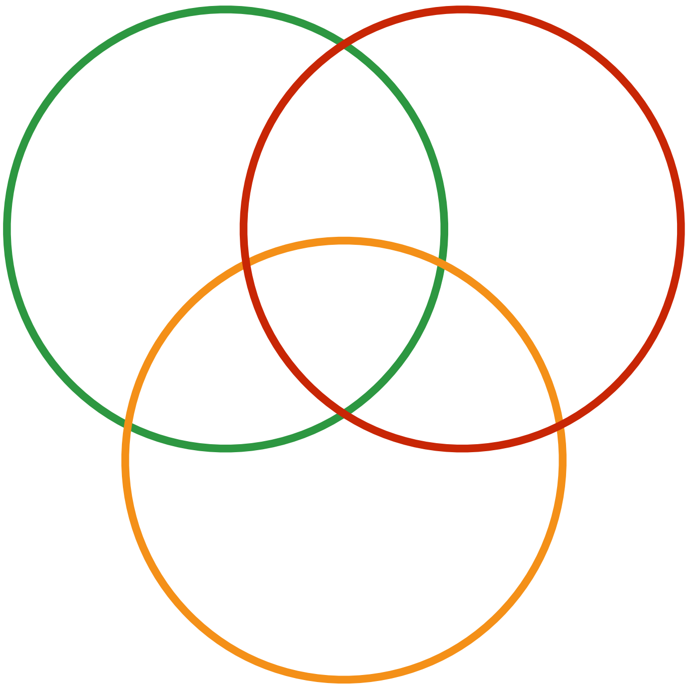
$$\{x \mid G(x) \oplus R(x)\}$$

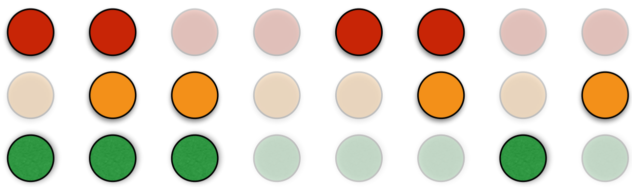
R	A	G	**G ⊕ R**
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**



xor

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

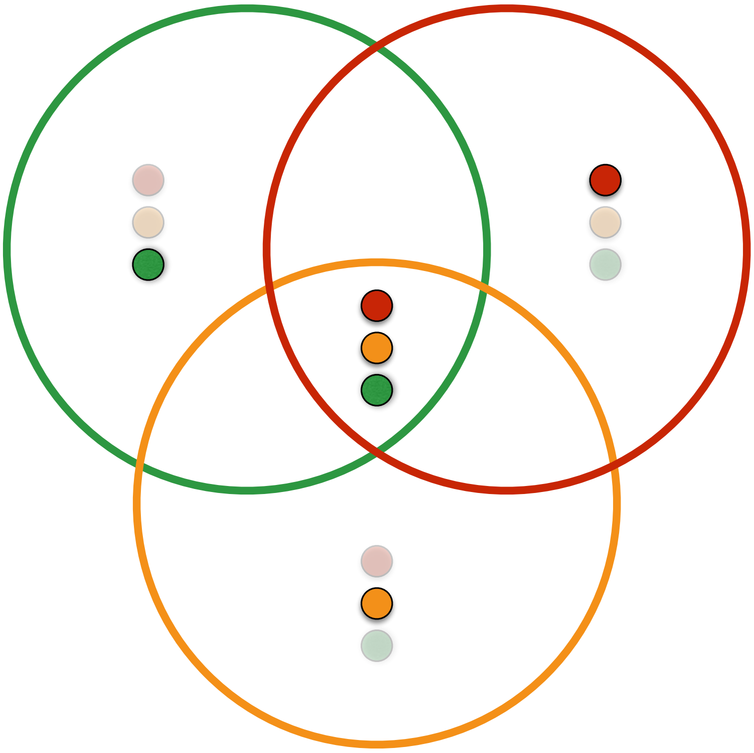


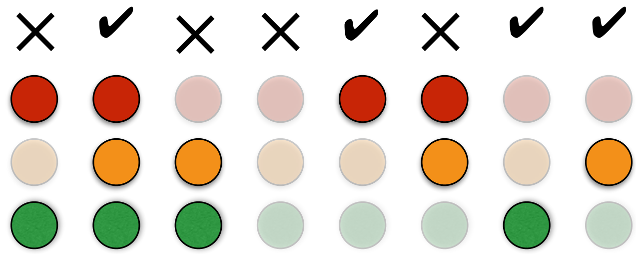


??

xor

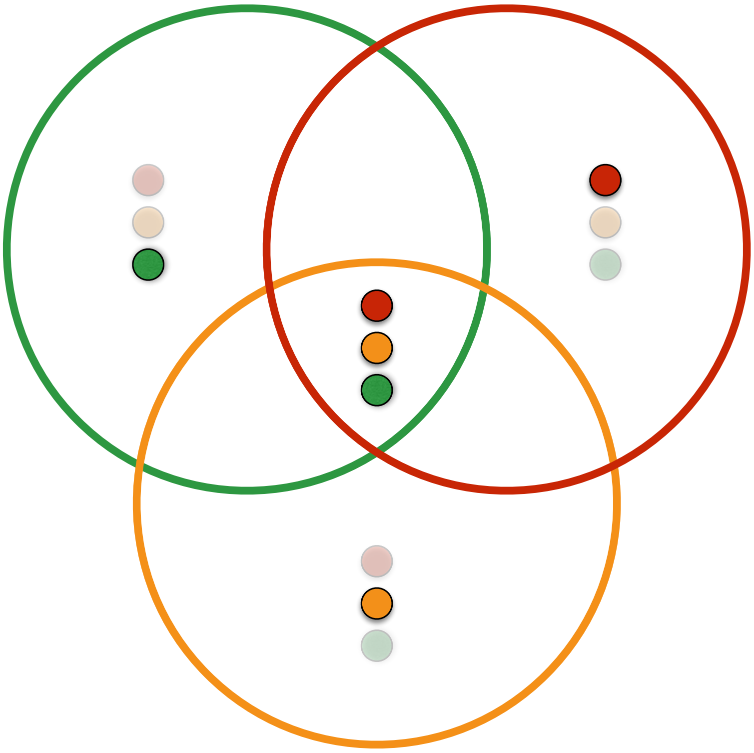
$$\{x \mid G(x) \oplus (R(x) \oplus A(x))\}$$

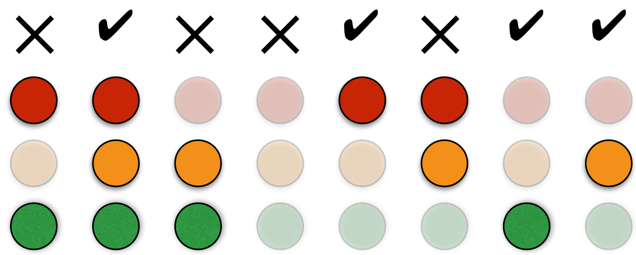




xor

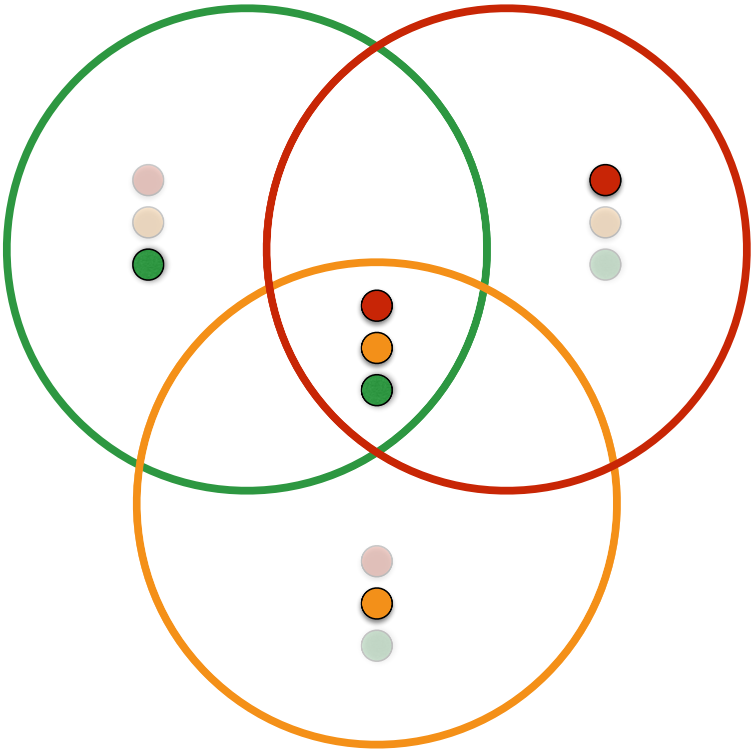
$$\{x \mid G(x) \oplus R(x) \oplus A(x)\}$$





xor

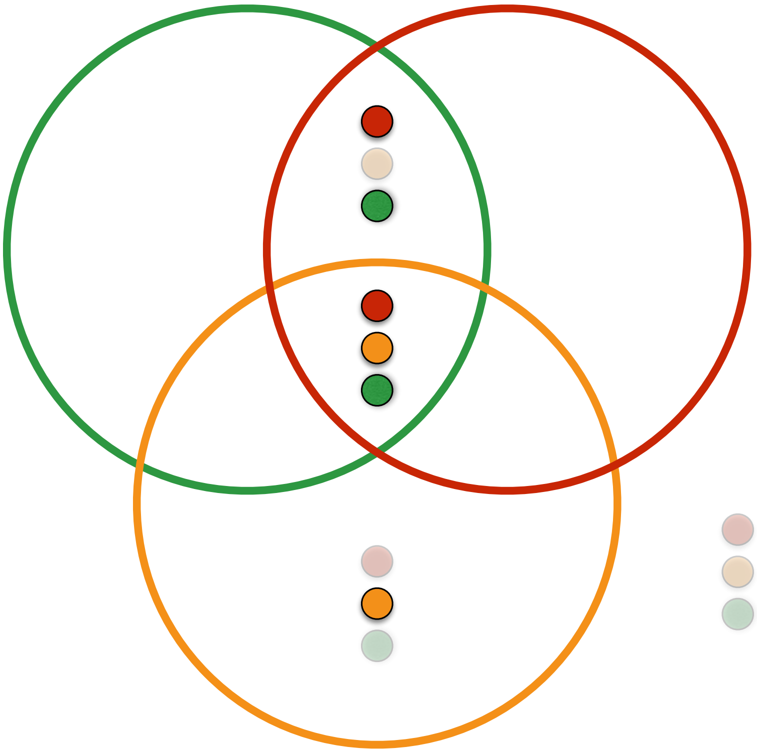
$$\{x \mid G(x) \oplus R(x) \oplus A(x)\}$$

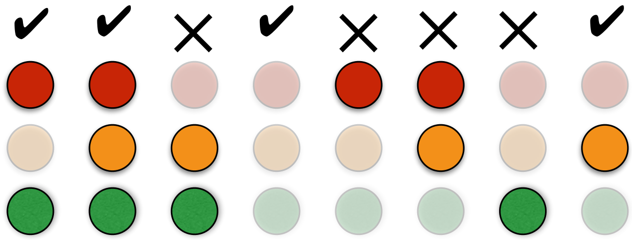


R	A	G	**R ⊕ A**	**R ⊕ A ⊕ G**
0	0	0	**0**	**0**
0	0	1	**0**	**1**
0	1	0	**1**	**1**
0	1	1	**1**	**0**
1	0	0	**1**	**1**
1	0	1	**1**	**0**
1	1	0	**0**	**0**
1	1	1	**0**	**1**

iff

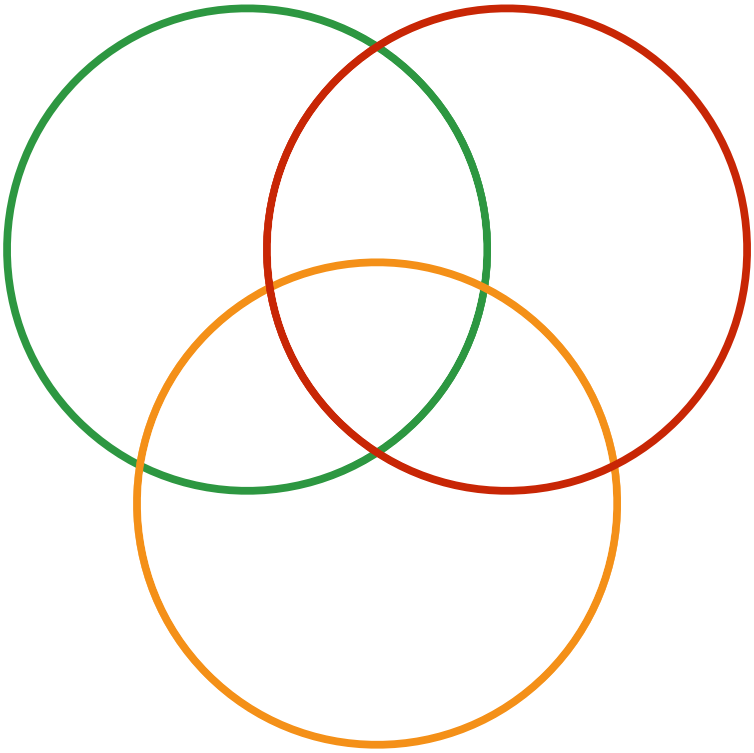
$$\{x \mid G(x) \leftrightarrow R(x)\}$$

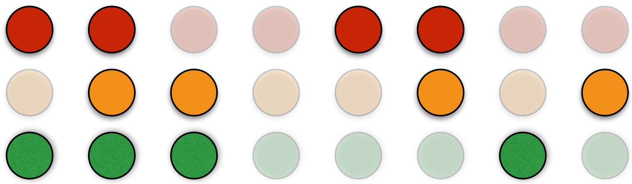




iff

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

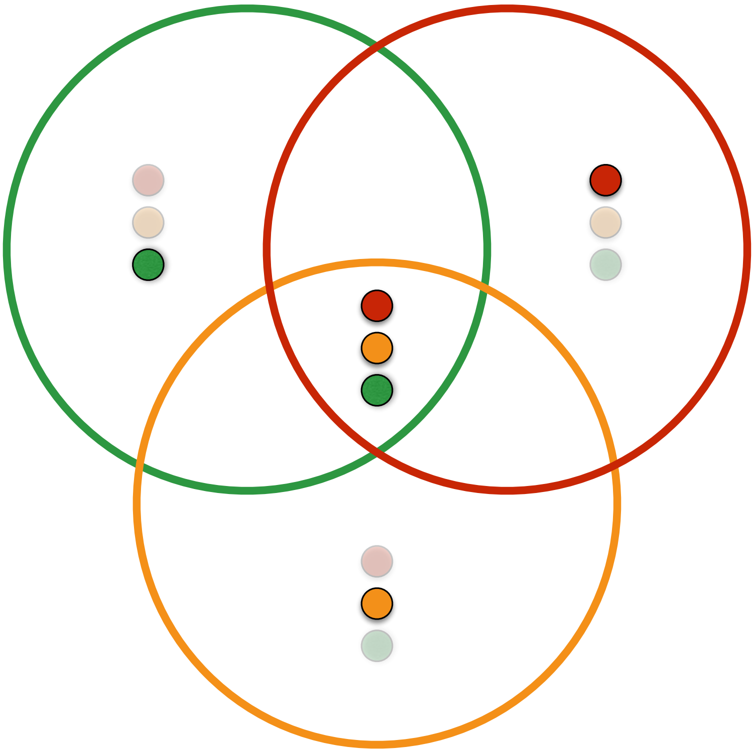


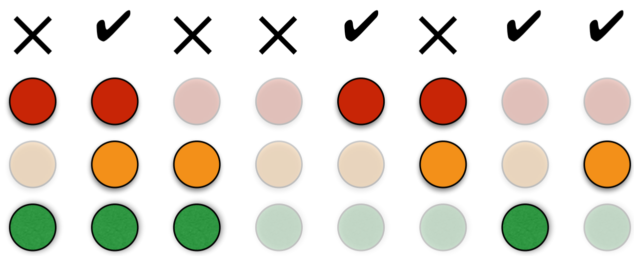


??

iff

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

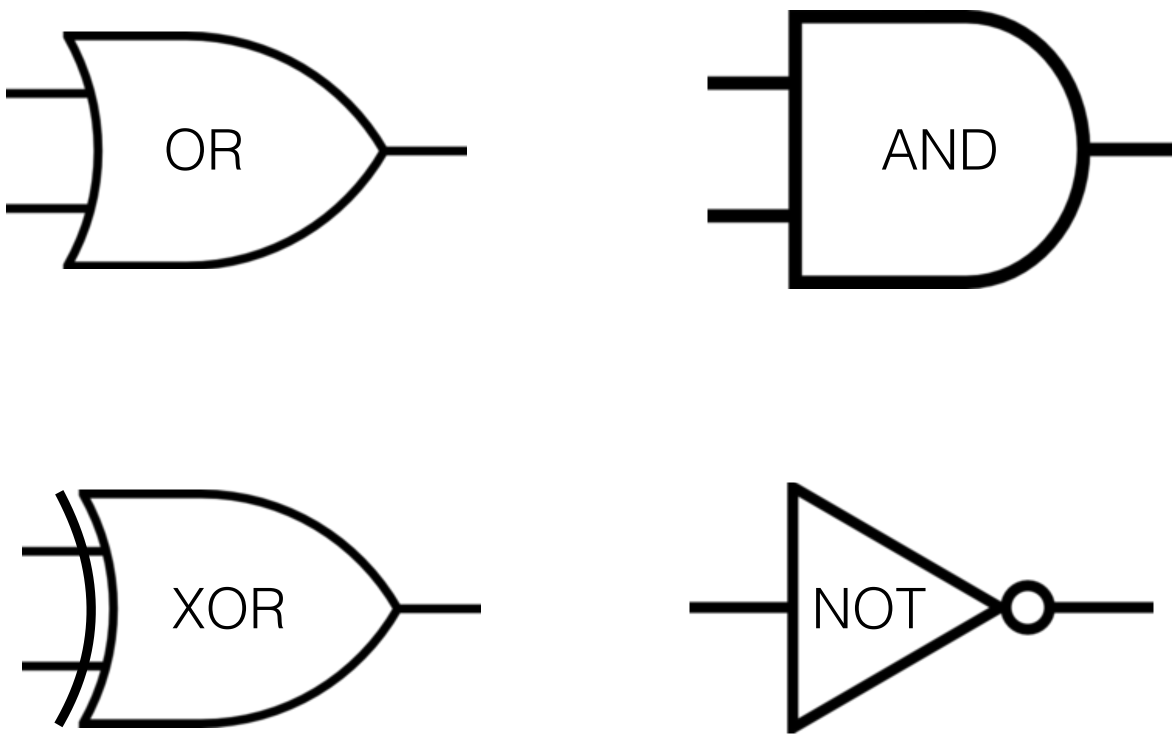


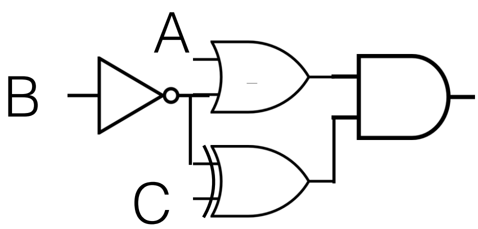


$$G(x) \leftrightarrow R(x) \leftrightarrow A(x)$$

$$\equiv$$

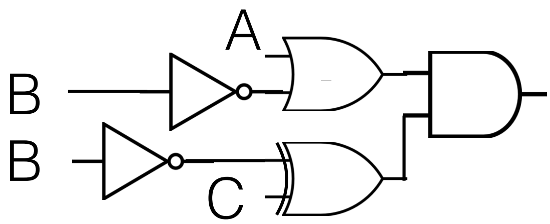
$$G(x) \oplus R(x) \oplus A(x)$$


OR
AND
XOR
NOT




```
let bbar = not B
    in (A || bbar) && (bbar ⊕ C)
```

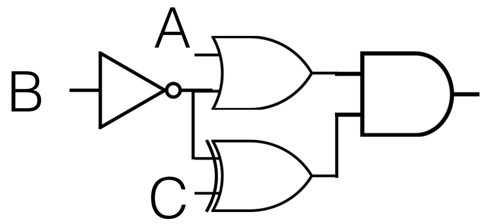
$$(A \vee \neg B) \wedge (\neg B \oplus C)$$




```
(A || not B) && (not B ⊕ C)
```

Exercise: define ⊕ in Haskell

Circuit

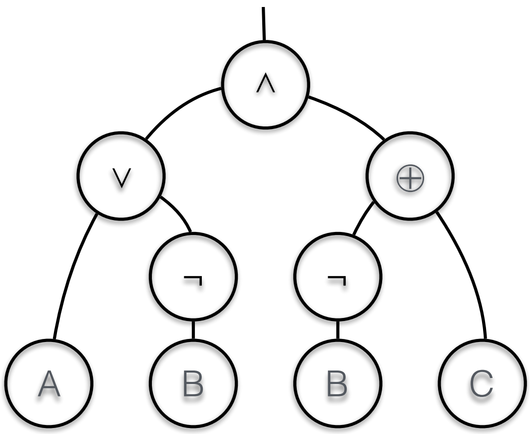


Formula

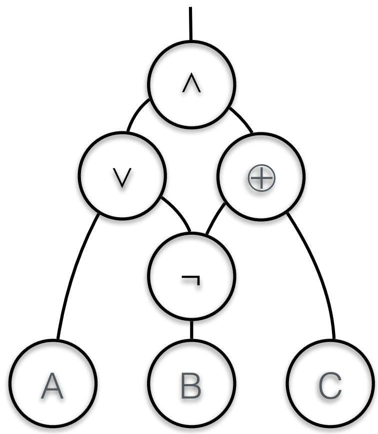
$(A \lor \neg B) \land (\neg B \oplus C)$

ABC	A∨¬B	¬B⊕C	out
000	1	1	1
001	1	0	0
010	0	0	0
011	0	1	0
100	1	1	1
101	1	0	0
110	1	0	0
111	1	1	1

Function

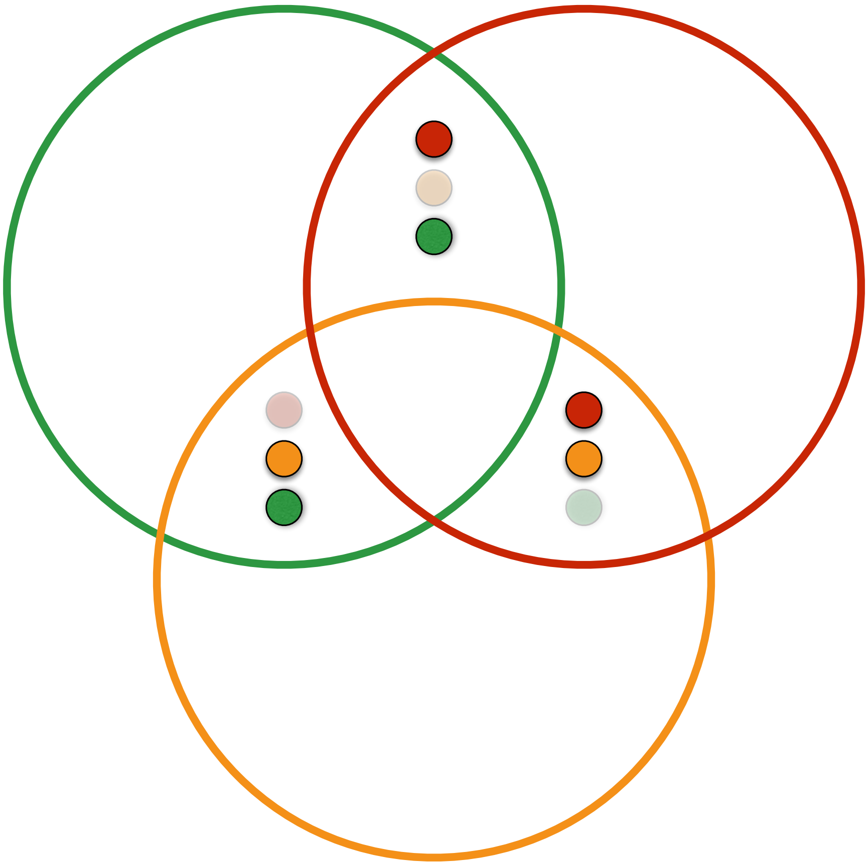


Syntax tree



DAG

Find a proposition



??

Basic Boolean operations

Boole (1815 – 1864)

$\mathbf{1}, \top$	true, top
$\vee$	disjunction, or
$\wedge$	conjunction, and
$\neg$	negation, not
$\mathbf{0}, \bot$	false, bottom

$$\mathbb{Z}_2 = \{0, 1\}$$

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

–	
0	0
1	1

$$x \wedge y \equiv xy$$

$$x \vee y \equiv x + y - xy$$

$$\neg x \equiv 1 - x$$

Here, we use arithmetic mod 2
The same equations work if we use ordinary arithmetic!

∨	0	1
0	0	1
1	1	1

∧	0	1
0	0	0
1	0	1

¬	
0	1
1	0

The algebra of sets

$$\mathcal{P}(S) = \{X \mid X \subseteq S\}$$

$X \vee Y = X \cup Y$	union
$X \wedge Y = X \cap Y$	intersection
$\neg X = S \setminus Y$	complement
$0 = \emptyset$	empty set
$1 = S$	entire set

Derived Operations

Definitions:

$$x \to y \equiv \neg x \vee y \qquad \text{implication}$$

$$x \leftarrow y \equiv x \vee \neg y$$

$$x \leftrightarrow y \equiv (\neg x \wedge \neg y) \vee (x \wedge y) \qquad \text{equality (iff)}$$

$$x \oplus y \equiv (\neg x \wedge y) \vee (x \wedge \neg y) \qquad \text{inequality (xor)}$$

Some equations:

$$x \leftrightarrow y = (x \to y) \wedge (x \leftarrow y)$$

$$x \oplus y = \neg(x \leftrightarrow y)$$

$$x \oplus y = \neg x \oplus \neg y$$

$$x \leftrightarrow y = \neg(x \oplus y)$$

$$x \leftrightarrow y = \neg x \leftrightarrow \neg y$$