INFORMATICS CLASS REPS

Get in touch with us!
We're here to help you - so feel free to approach us directly.

Harjyot (Harry) – 07831002358 / harjyot7@gmail.com or facebook.com/Harjyot

Liam - 07834476281 / s1440672@sms.ed.ac.uk

Lisa - 07961467009 / lxie.ger@gmail.com

Feel free to bring up issues such as:

- **Deadlines**
- **Course objectives**
- **Timings or other issues with tutorials or lectures**
- **Any feedback about the delivery, content, assessment, feedback and materials for your courses**

WE DON'T BITE!

Informatics 1

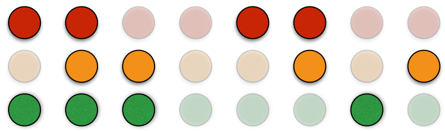
Computation and Logic

Sets of States: Venn Diagrams and Truth Tables

Michael Fourman

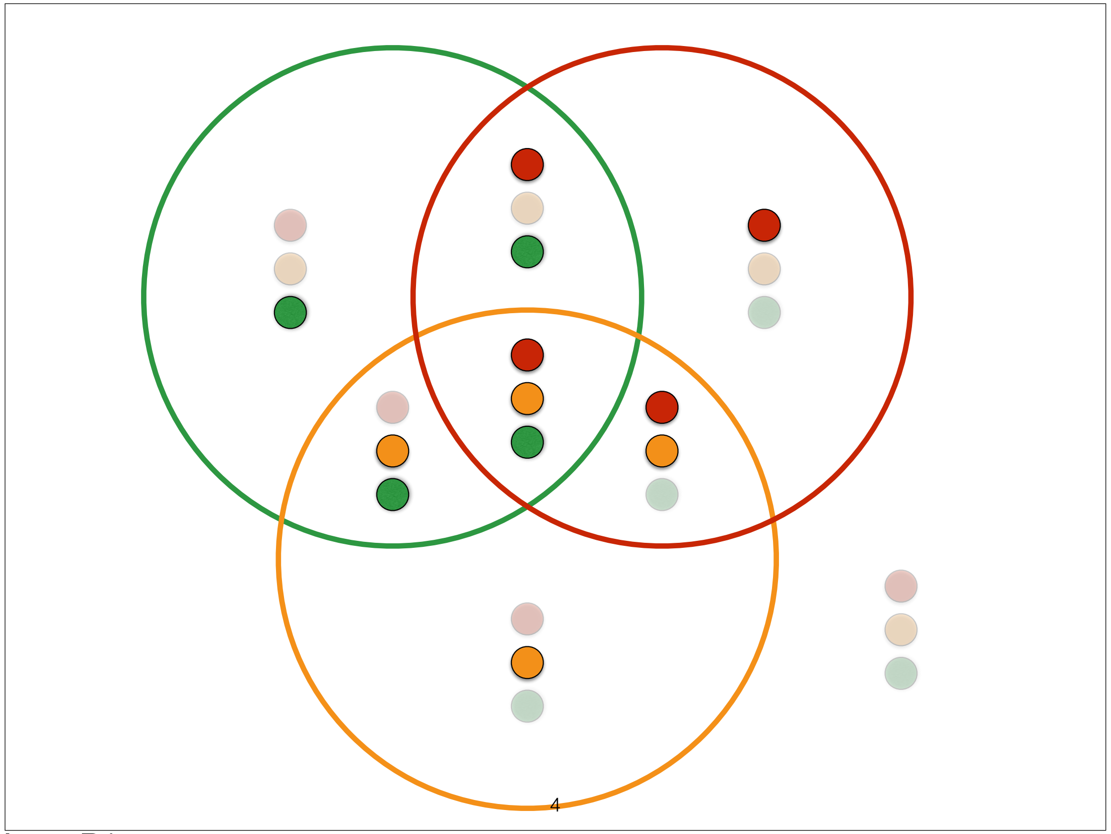
This course provides a first glimpse of the deep connections between computation and logic. We will focus primarily on the simplest non-trivial examples of logic and computation: propositional logic and finite-state machines.

In this first lecture we look at an example that introduces some ideas that we will explore further in later lectures, and introduce some notation which should become more familiar in due course.



The states of the signal

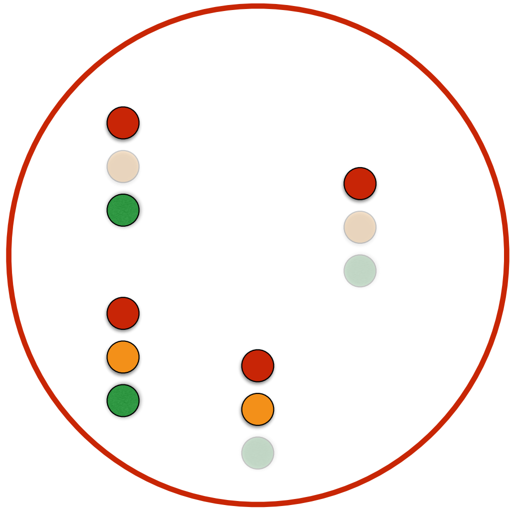
We are going to look at sets of states - all 256 of them



We can place the states in a Venn Diagram.

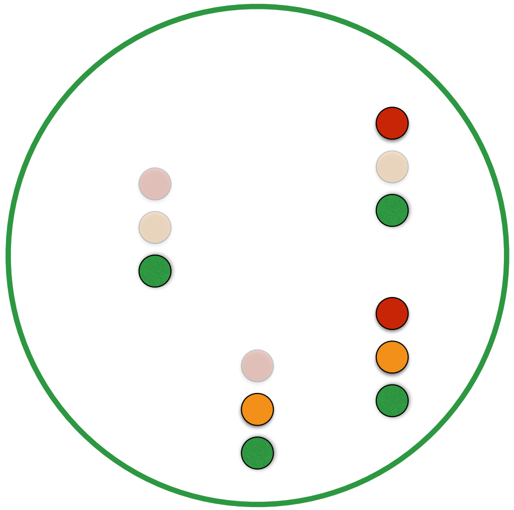
This includes all eight possible combinations of values for the three Boolean state variables.

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

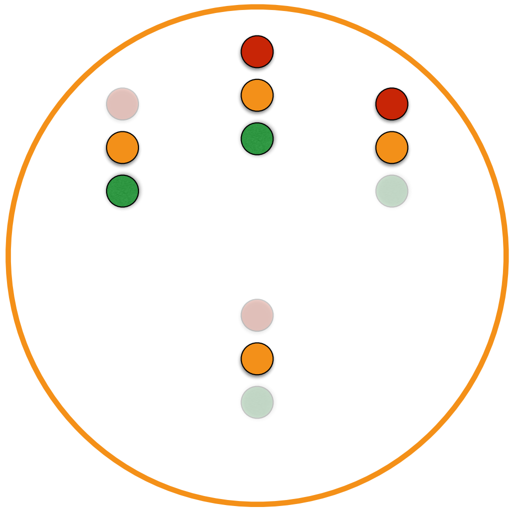


This notation for *set comprehension* will be useful.

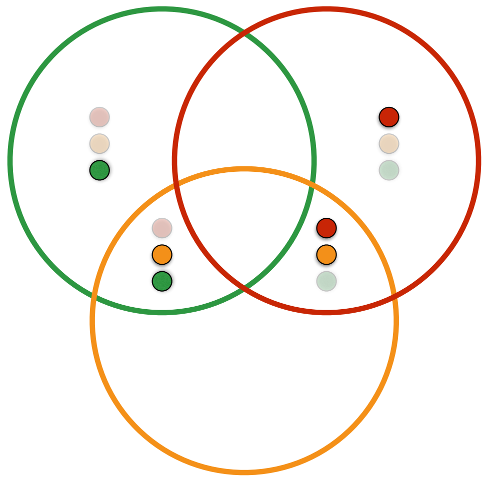
$$\{x \mid G(x)\}$$



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

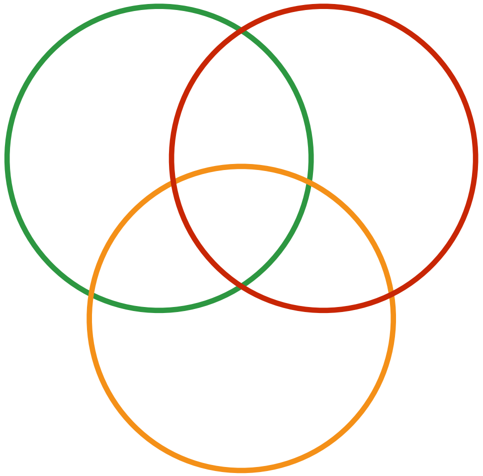


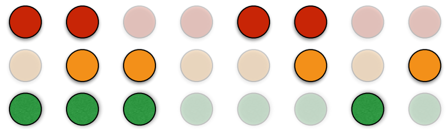
$$\{x \mid G(x) \oplus R(x)\}$$



We can compute the set of states corresponding to any expression

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

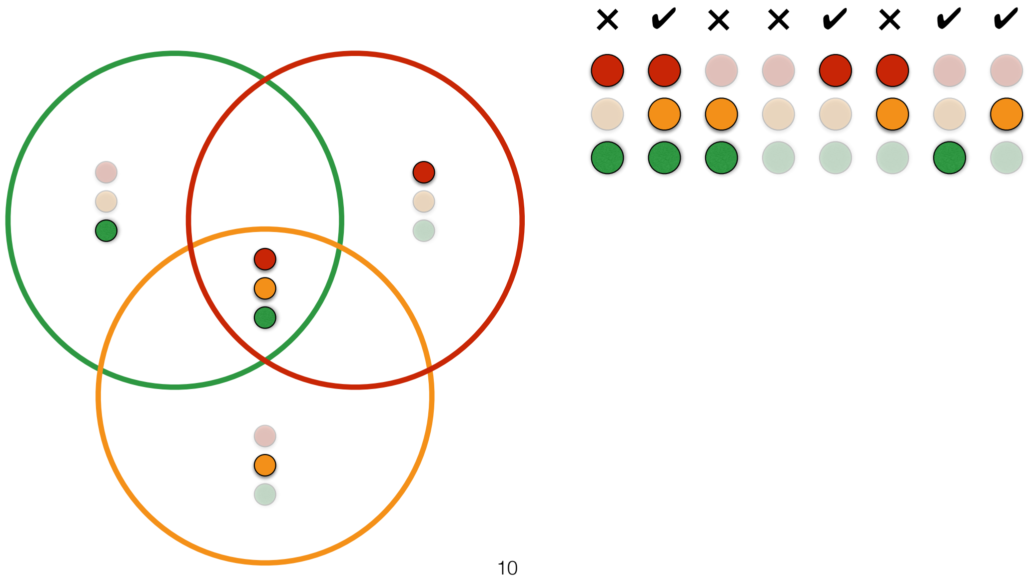




??

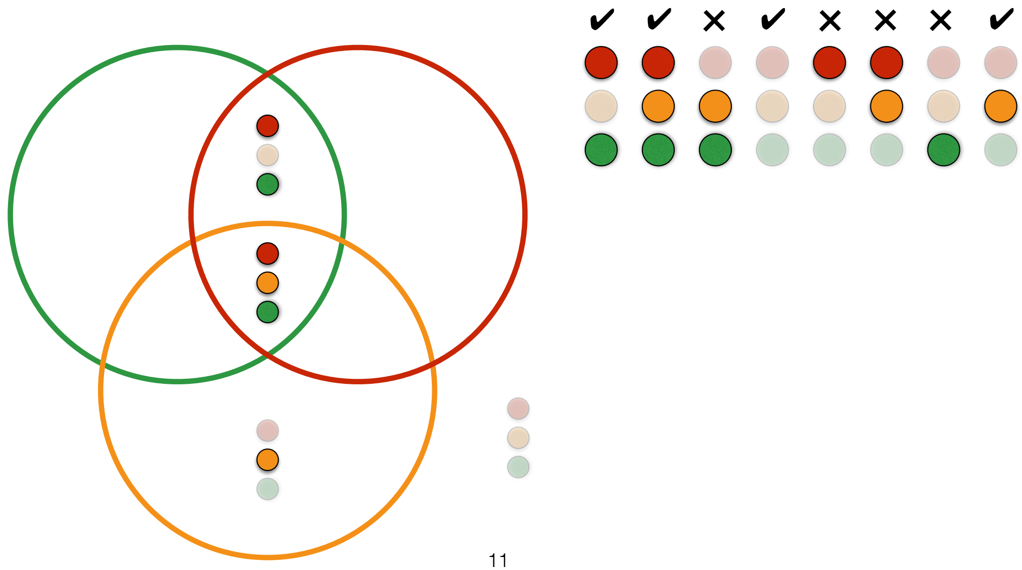
To try in class

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

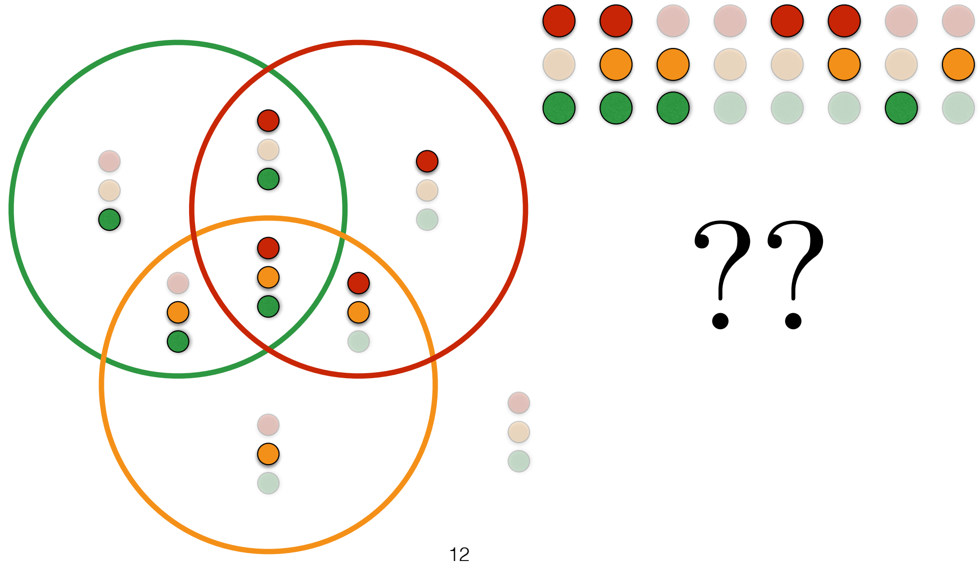


We find that the solution is symmetric, so *xor* is associative.

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

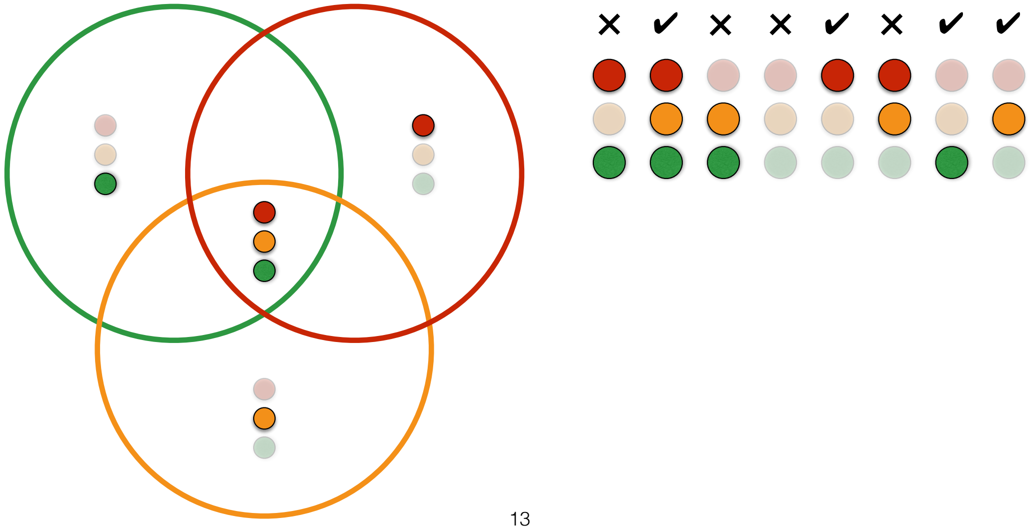


$$\{x \mid G(x) \leftrightarrow R(x) \leftrightarrow A(x)\}$$



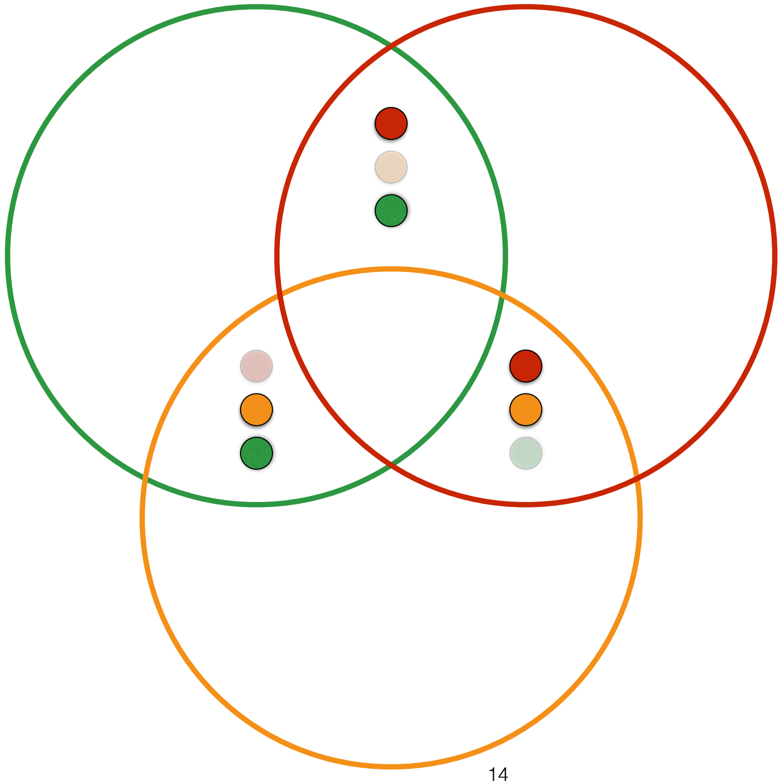
To do in class

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



To determine whether to expressions are equivalent, we can check whether they give the same values for all 2^n states of the system

Venn diagram is just a presentation of truth table for two or three variables.



??

Basic Boolean operations

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

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

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

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

Definitions:

$$
\begin{aligned}
x \wedge y &\equiv xy \\
x \vee y &\equiv x + y - xy \\
\neg x &\equiv 1 - x
\end{aligned}
$$

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:

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

an algebraic proof

$$
\begin{aligned}
(x \leftrightarrow y) \leftrightarrow z &= \neg(x \leftrightarrow y) \leftrightarrow \neg z && (a \leftrightarrow b = \neg a \leftrightarrow \neg b) \\
&= (x \oplus y) \leftrightarrow \neg z && (\neg(a \leftrightarrow b) = a \oplus b) \\
&= (x \oplus y) \oplus z && (a \leftrightarrow \neg b = a \oplus b)
\end{aligned}
$$

Once we know the rules for iff and xor shown on the right, we can give an algebraic proof that the xor combination of three variables is the same as their iff combination

Boolean connectives

Some equalities:

$$x \vee y = \neg(\neg x \wedge \neg y) \qquad x \wedge y = \neg(\neg x \vee \neg y)$$
$$\neg x = x \to 0 \qquad x \vee y = \neg x \to y$$

We will see that $\wedge$, $\vee$, $\neg$ and $\perp$ are sufficient to define any boolean function. These equations show that $\{\wedge, \neg, \perp\}$, $\{\vee, \neg, \perp\}$, and $\{\to, \perp\}$ are all sufficient sets.

Boolean Algebra

$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	associative
$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	distributive
$x \vee y = y \vee x$	$x \wedge y = y \wedge x$	commutative
$x \vee 0 = x$	$x \wedge 1 = x$	identity
$x \vee 1 = 1$	$x \wedge 0 = 0$	annihilation
$x \vee x = x$	$x \wedge x = x$	idempotent
$x \vee \neg x = 1$	$\neg x \wedge x = 0$	complements

$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$	absorbtion
$\neg(x \vee y) = \neg x \wedge \neg y$	$\neg(x \wedge y) = \neg x \vee \neg y$	de Morgan
$\neg\neg x = x$	$x \rightarrow y = \neg x \leftarrow \neg y$	

The equations above the gap define a Boolean algebra.

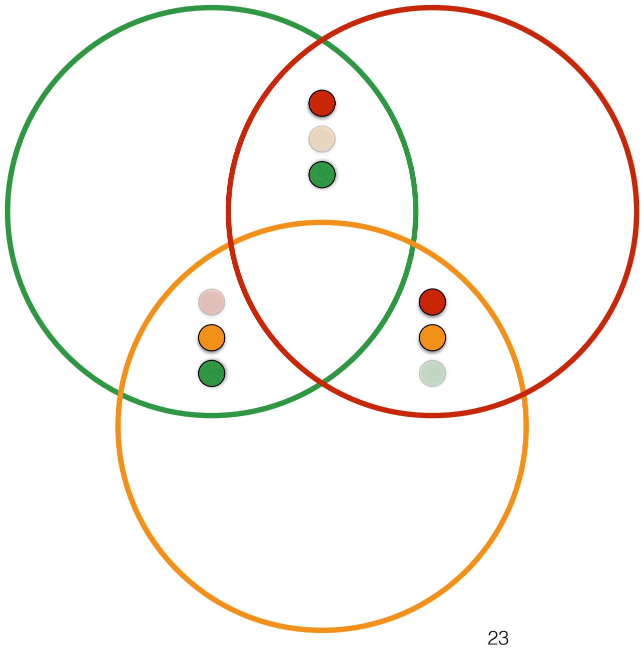
Those below the line follow from these.

Exercise 2.1

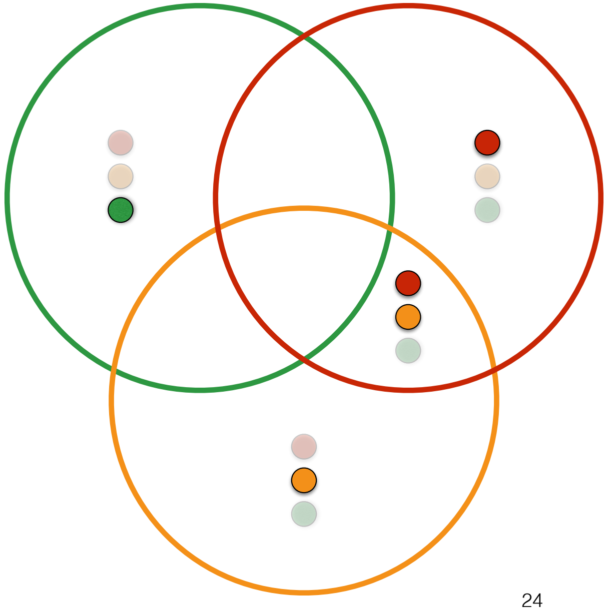
Which of the following rules are *not* valid for arithmetic?

$x + (y + z) = (x + y) + z$	$x \times (y \times z) = (x \times y) \times z$	associative
$x + (y \times z) = (x + y) \times (x + z)$	$x \times (y + z) = (x \times y) + (x \times z)$	distributive
$x + y = y + x$	$x \times y = y \times x$	commutative
$x + 0 = x$	$x \times 1 = x$	identity
$x + 1 = 1$	$x \times 0 = x$	annihilation
$x + x = x$	$x \times x = x$	idempotent
$x + (x \times y) = x$	$x + (x \times y) = x$	absorbtion
$x + -x = 1$	$x \times -x = 0$	complements

Exercise 2.2 Generate CNF for this subset



Exercise 2.3 Generate CNF for this subset



Exercise 2.4 (for mathematicians)

In any Boolean algebra, define,

$$x \leq y \equiv x \wedge y = x$$

1. Show that, for any x, y, and z,

$$\begin{gathered} 0 \leq x \text{ and } x \leq x \text{ and } x \leq 1 \\ x \to y = \top \ \text{ iff } x \leq y \\ \text{if } x \leq y \text{ and } y \leq z \text{ then } x \leq z \\ \text{if } x \leq y \text{ and } y \leq x \text{ then } x = y \\ \text{if } x \leq y \text{ then } \neg y \leq \neg x \end{gathered}$$

2. Show that, in any Boolean Algebra,

$$x \wedge y = x \text{ iff } x \vee y = y$$