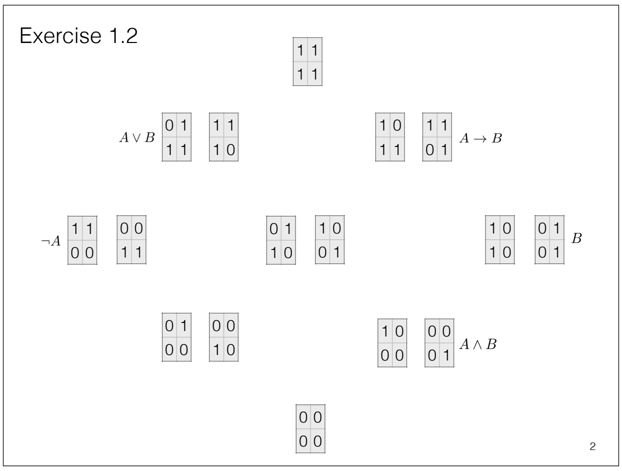
Informatics 1

Computation and Logic

Entailment

Michael Fourman



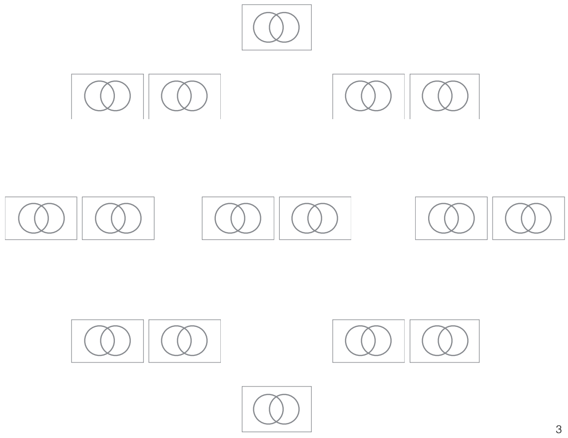
Each of the 16 2x2 tables above represents the truth table of a binary boolean operation.

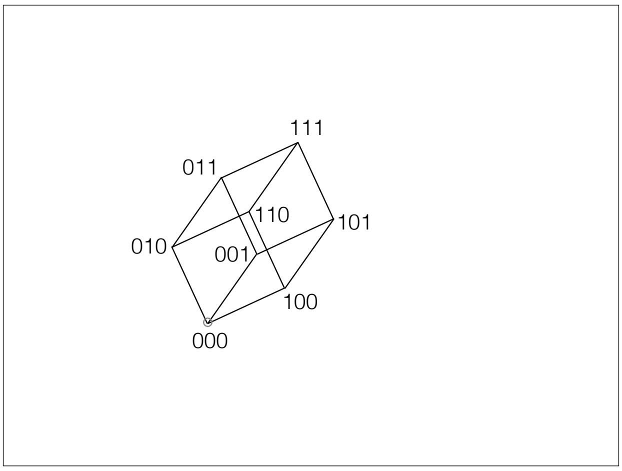
Label each table with a boolean expression for which it is the truth table (five tables are already labelled – begin by checking whether these are consistent).

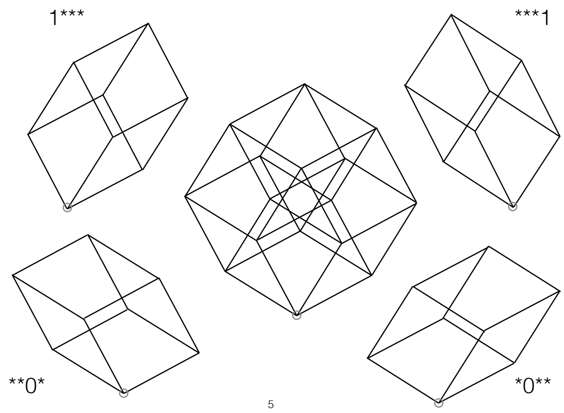
How many of the binary operations actually depend on both variables?

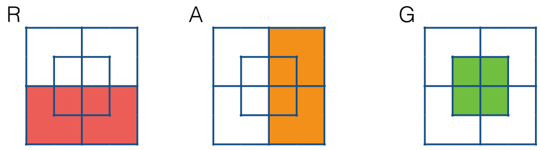
How many depend on only one variable?

How many depend on no variables?

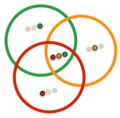


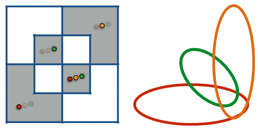

111
011
110
101
010
001
100
000


1***
***1
**0*
*0**



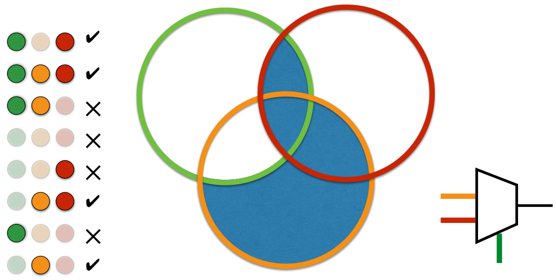
$$\{x \mid G(x) \leftrightarrow R(x) \leftrightarrow A(x)\}$$



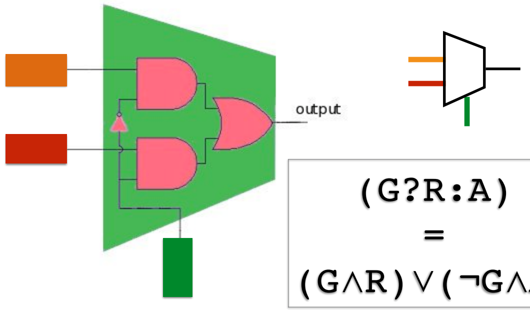


(G?R:A)

if G then R else A



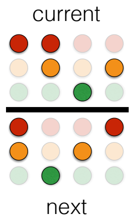
multiplexer – ITE



$$(G?R:A) = (G \wedge R) \vee (\neg G \wedge A)$$

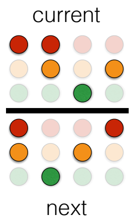
`(R?    :      )`

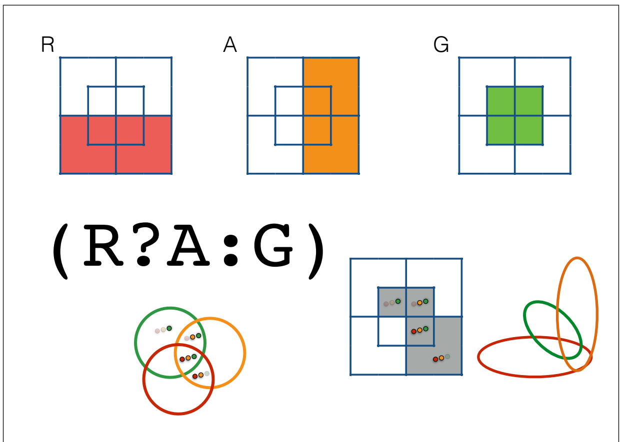
legal states

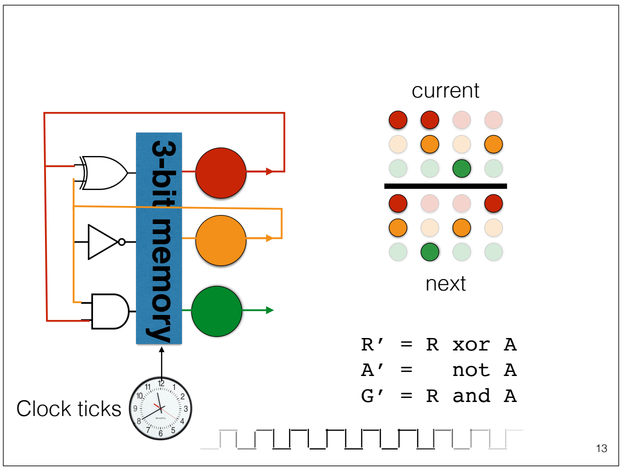


$$(\mathtt{R?}\neg\mathtt{G:A}\oplus\mathtt{G})$$

legal states



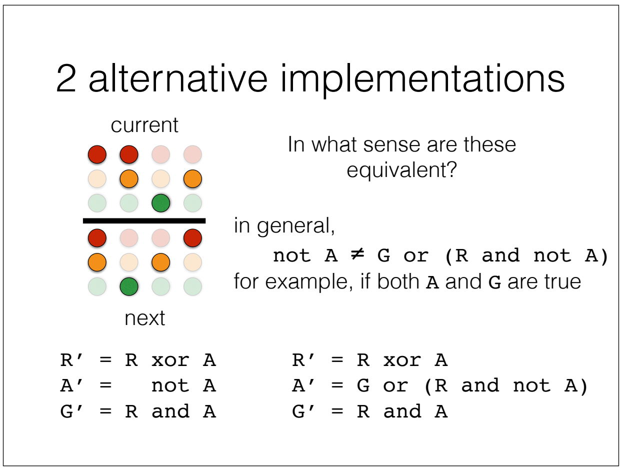

R
A
G
(R?A:G)



The next-state logic for sequencing our traffic lights can be implemented using three different gates. Many different technologies can be used to implement logic gates, some may use high and low voltages to represent binary values, others might use currents, but this logical description of our circuit provides a common abstract level of design.

In our diagram, the current state is shown in the three coloured discs. The outputs of the three gates represent the next state. To make the state transition we need to replace the current state by the next state.

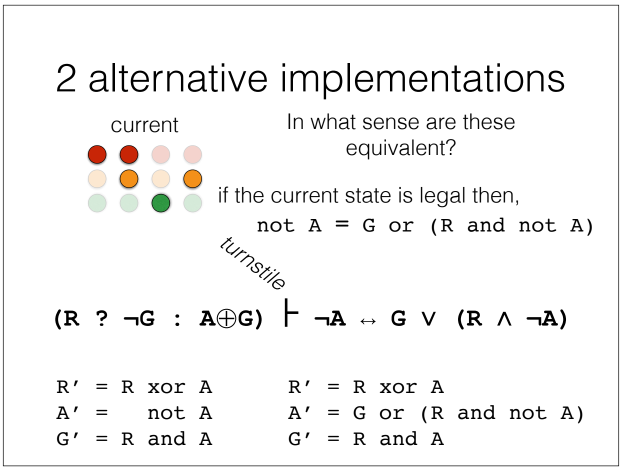
We need memory. One simple form of memory is a *latch*, a special kind of circuit with two inputs, *data* and *clock*. When the clock ticks the current input data value is loaded and stored. The stored value is output, and does not change until the next tick of the clock.



We have two implementations of the controller, with different next-state logic for the amber light, A.

A′ = not A

A′ = G or (R and not A).



Slide 25 (lecture 1) shows an implementation of the traffic light controller.

We could have designed our logic differently.

For example, letting

A′ = G or (R and not A).

Draw the circuit for this implementation.

Is this a correct implementation of the controller? Explain your answer.

Entailment

In algebra, we consider expressions with variables, and write equations to express relationships between different expressions.

LHS = RHS

Boolean algebra, with equalities between expressions, gives us one way to express relationships between different logical expressions.

If we want to study logical arguments it is more natural to consider entailments.

LHS ⊢ RHS

Entailment

If we want to study logical arguments it is more natural to consider entailments.

$$\text{LHS} \vDash \text{RHS}$$

The entailment is valid if any valuation that makes everything on the LHS true, makes the RHS true

$$\vDash \text{RHS}$$

an entailment with empty LHS is valid iff RHS is a **tautology**
i.e. every valuation makes it true

Is this a valid argument?

- Assumptions:
  If I am clever then I will pass
  If I will pass then I am clever,
  Either I am clever or I will pass
- Conclusion:
  I am clever and I will pass

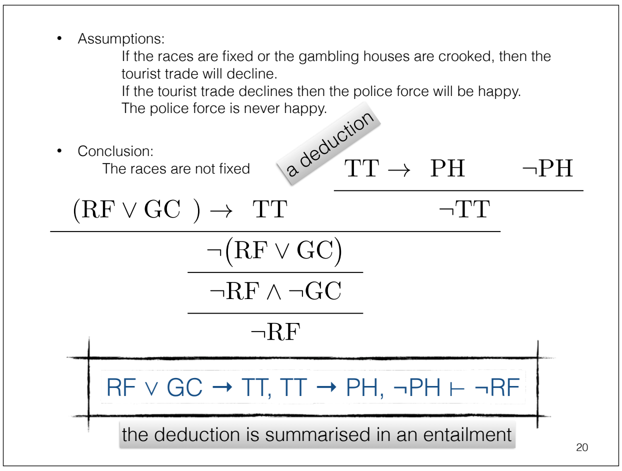
$$C \rightarrow P,\ P \rightarrow C,\ C \vee P \overset{?}{\vdash} C \wedge P$$

Use the abbreviations C (I am clever) and P (I will pass). We argue as follows: The first two assumptions tell us that $C \leftrightarrow P$, so the truth value of C is equal to the truth value of P. The third assumption tells us that at least one of them is true, so both must be.
(If we interpret `or' in the third assumption as xor, then the assumptions are inconsistent.)

Is this a valid argument?

- Assumptions:
  If the races are fixed or the gambling houses are crooked, then the tourist trade will decline.
  If the tourist trade declines then the police force will be happy.
  The police force is never happy.

- Conclusion:
  The races are not fixed



Here is a deduction, or proof. This sets out an argument, so that we can review how the steps fit together. Each line represents a simple step in the argument: we check that, if the propositions above the line are true then the conclusion below the line is true. The only propositions here that are not conclusions are our assumptions. So, if we assume the assumptions are true, then, working downwards, we see that each conclusion is true. The races are not fixed. The deduction shows that the entailment is valid.

RF The races are fixed.

GC The gambling houses are crooked.

TT The tourist trade declines.

PH The police force is happy.