Informatics 1 - Computation & Logic: Tutorial 1

Logic, States and Transitions

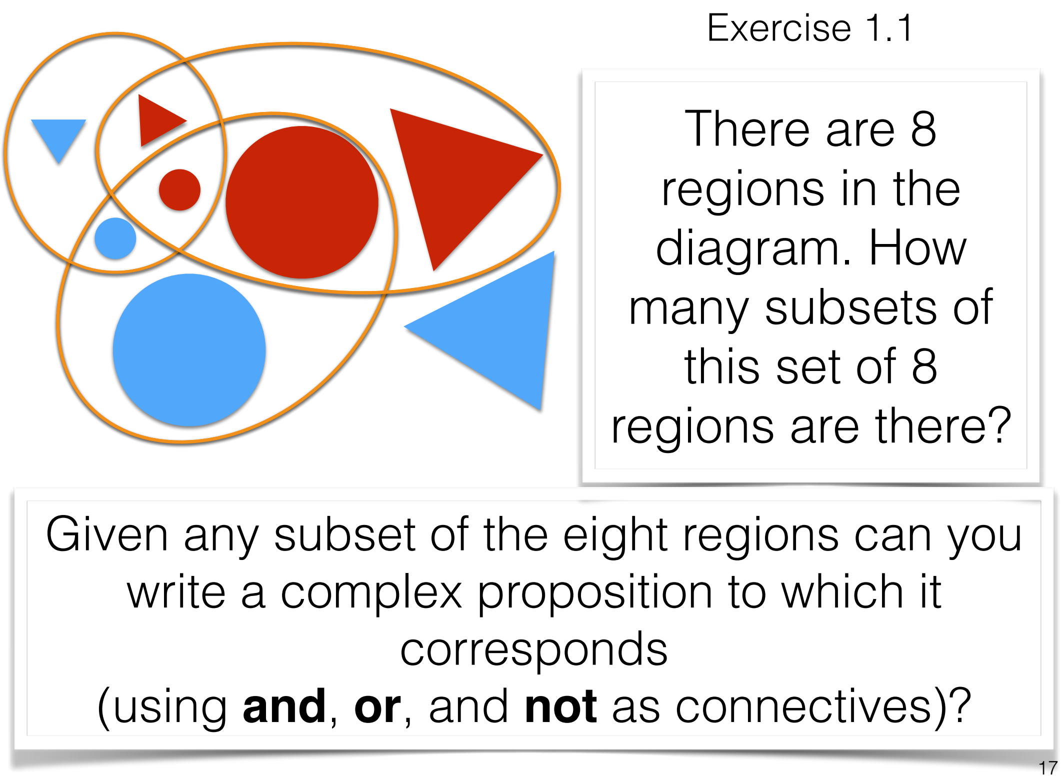
Week 3: 3–7 October 2016

Please attempt the entire worksheet in advance of the tutorial, and bring with you all work, including (if a computer is involved) printouts of code and test results. Tutorials cannot function properly unless you do the work in advance.

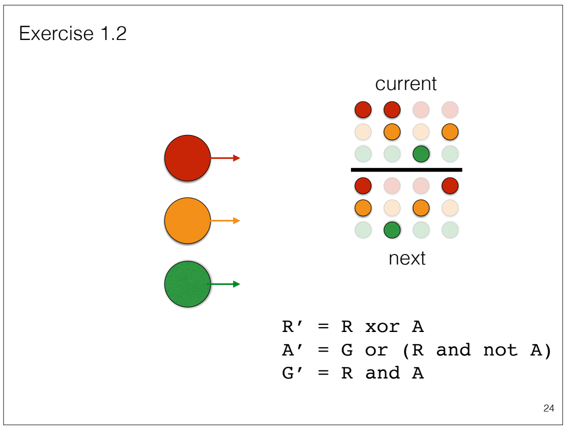
You may work with others, but you must understand the work; you can't phone a friend during the exam.

Assessment is formative, meaning that marks from coursework do not contribute to the final mark. But coursework is not optional. If you do not do the coursework you are unlikely to pass the exams.

Attendance at tutorials is **obligatory**; please let your tutor know if you cannot attend.

1. 

You may want to come back to this question after you have completed the rest of the tutorial.



2.

Slide 20 (of lecture 2) shows an implementation of the traffic light controller.

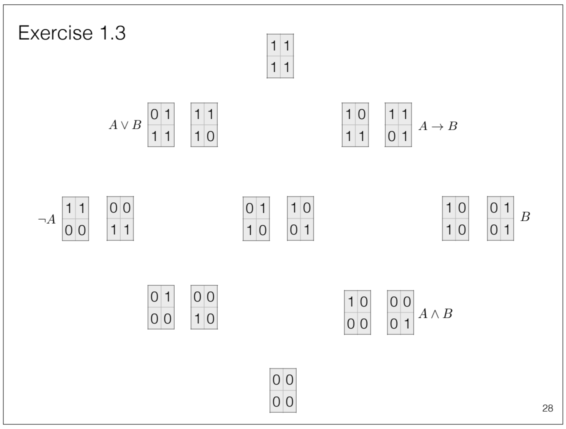
We could have designed our logic differently.

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.



3.

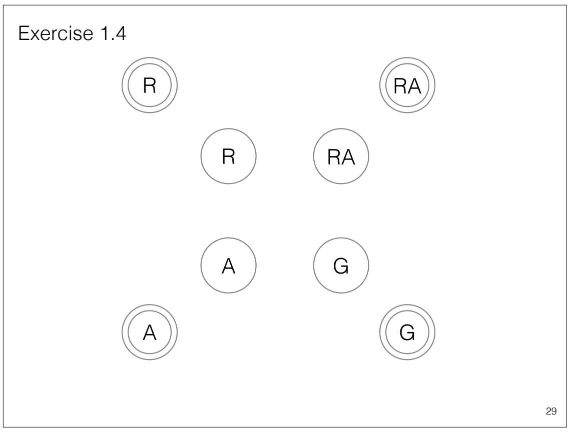
Each of the 16 2x2 tables above represents the truth table of a binary boolean operation.

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 correct).

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

How many depend on only one variable?

How many depend on no variables?



4.

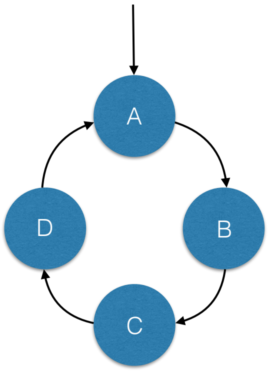
As discussed in the lecture, the diagram represents the beginnings of a refinement of our description of the traffic light controller. We model a sensor that detects a car ready to pass the light. For each state of the lights, (R, RA, G, A) we have two states, one (with a double circle) where there is a car, and the other, without a car, as before.

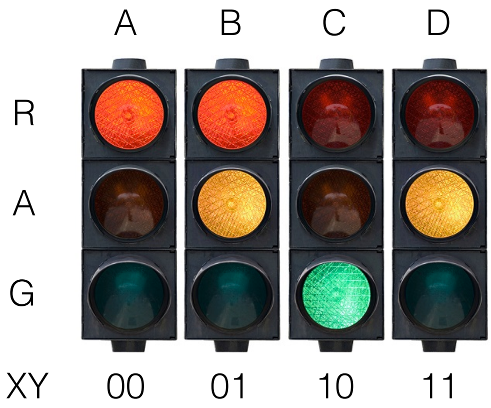
Draw arrows to indicate state changes that still obey the correct sequence for the lights, but also respect the following two rules.

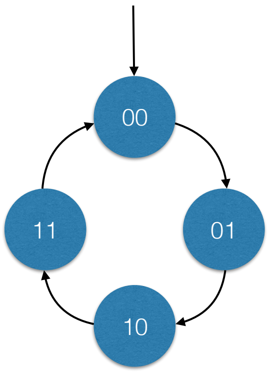
1. A car can only pass the light if it is green.
2. The light only changes from red to red-amber when a car is detected

Optional: You may also design the logic for the controller.

Use a new boolean variable C to represent the presence of a car, and give equations for R′ A′ and G′. Should we also give an equation for C′ ?







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

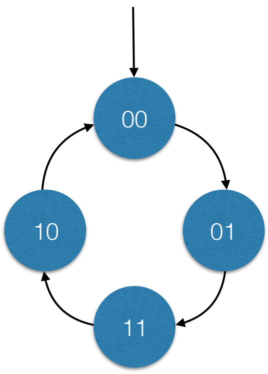
$$X' = X \oplus Y \text{ and } Y' = \neg Y$$

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

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

5.

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

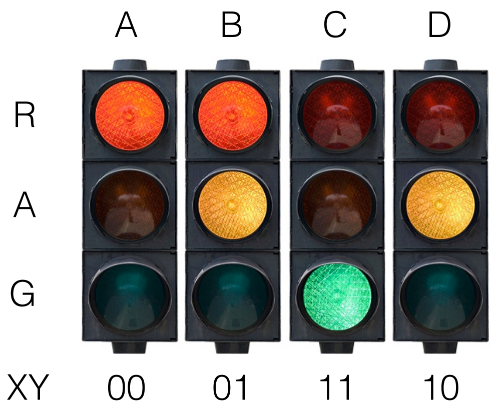


Give expressions for the next state logic

$X' =$ $Y' =$

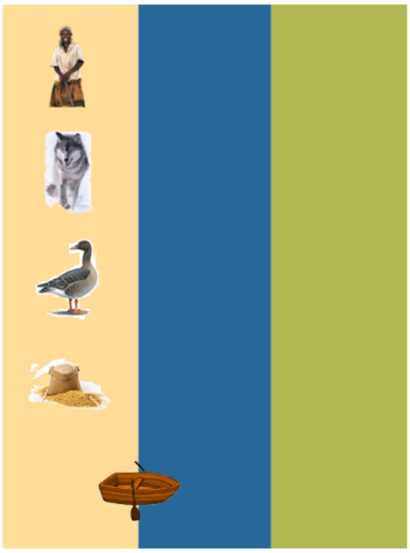
and the output logic

$R =$ $A =$ $G =$



6.

www.inf.ed.ac.uk/teaching/courses/inf1/cl/FWC/



How can we use logic to specify the transitions?

This is a **non-deterministic** system. We define a next state **relation**.

Again we introduce next state variables WW' etc.

Here we have
$FW \wedge WW \wedge GW \wedge CW$

Is it possible that WE' ? **NO**

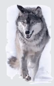
One thing true in our model is that
$WE' \rightarrow WE \vee WB$

What else do we need to say to give a complete description ?

What does it mean for a description to be complete?

29

7. We used four conditions to define the state space for the Wolf, Goose, Corn puzzle, using 12 atomic propositions. This question asks you to do the same using a different set of atoms.

	West		East
	WW	WB	WE
	CW	CB	CE
	GW	GB	GE
	FW	FB	FE

one place	(WW ⊕ WB ⊕ WE) ∧ ¬(WW ∧ WB ∧ WE)	×4 (wolf,goose,corn,farmer)
not solo	GB → FB	×3 (wolf,goose,corn)
no conflict	GW ∧ (WW ∨ CW) → FW	×2 (east, west)
no overload	¬(GB ∧ CB) ∧ ¬(GB ∧ WB) ¬(WB ∧ CB)	×1

9

	West	East
	~~WW~~	~~WE~~
	~~CW~~	~~CE~~
	~~GW~~	~~GE~~
	~~FW~~	~~FE~~

We can use different atoms to model the system.

We introduce 8 atoms whose meanings are given as the negations of the east and west propositions we used earlier.

e.g ~~WW~~ ↔ ¬WW

We can define the old propositions in terms of the new ones:

WB ↔ (~~WW~~ ∧ ~~WE~~) ×4

WW ↔ ¬~~WW~~ ×8

This encoding uses only 8 propositional atoms – 256 states.

one place	~~WW~~ ∨ ~~WE~~	×4
not solo		×3
no conflict		×2
no overload		×1

The oneplace axiom is now simpler. Each instance eliminates one quarter of the remaining states, leaving 192 , 144, 108, and 81

Give the other axioms in terms of the new atoms.
How many states are eliminated by each of your axioms?

11

8. Give expressions defining the next-state relation for the Wolf, Goose, Corn puzzle, in terms of the new atoms.

Truth tables of the basic operators

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

A	B	$A \to B$
T	T	T
T	F	F
F	T	T
F	F	T

We can also express the truth tables for binary operations in a different style, which makes the symmetries more immediately apparent.

$x \wedge y$		y = 0	y = 1
x	0	0	0
	1	0	1

$x \vee y$		y = 0	y = 1
x	0	0	1
	1	1	1

$x \to y$		y = 0	y = 1
x	0	1	1
	1	0	1

$x \leftrightarrow y$		y = 0	y = 1
x	0	1	0
	1	0	1

$x \oplus y$		y = 0	y = 1
x	0	0	1
	1	1	0

This tutorial exercise sheet was written by Michael Fourman.
Please send comments to `michael.fourman@ed.ac.uk`