

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.

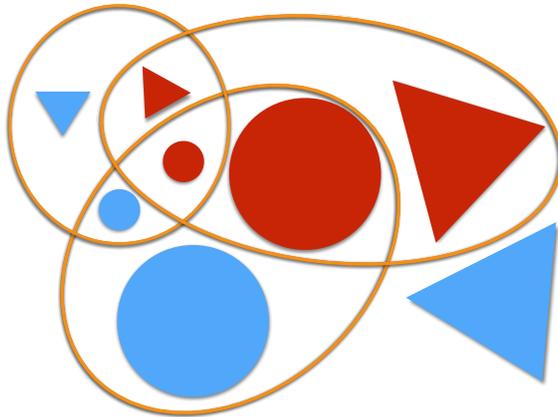
You may work with others, but you must understand the work; you can't phone a friend during the exam.

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.

Exercise 1.1

1.



There are 8 regions in the diagram. How many subsets of this set of 8 regions are there?

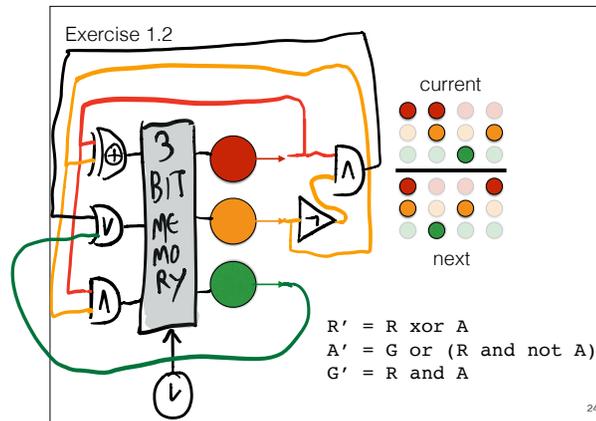
Given any subset of the eight regions can you write a complex proposition to which it corresponds (using **and**, **or**, and **not** as connectives)?

17

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

To describe an arbitrary subset you must, for each region, specify whether it is included or not; thus, there are two possibilities for each region, and eight regions. Hence, there are $2^8 = 256$ subsets.

The second part of the question can be answered by characterising each region in the subset by a conjunction of literals, and then taking the disjunction of these. (There are also other ways.)



2.

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

We could have designed our logic differently.

For example, letting

$A' = G \text{ or } (R \text{ and not } A)$.

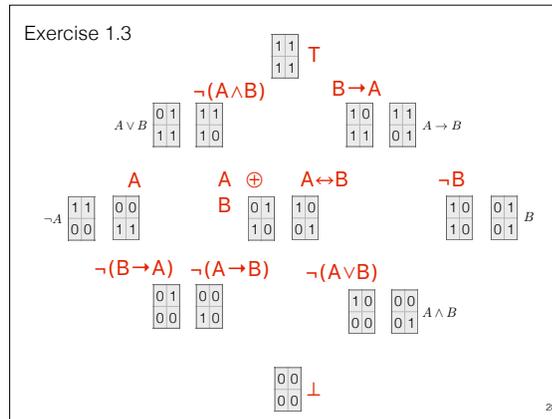
Draw the circuit for this implementation.

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

This implementation is based on a logical characterisation, for each light, of the set of current states such that the light is on in the next state. Check that the expression for A' gives the correct answer in each of the four cases.

In tutorial you should discuss the fact that the expression for A' is NOT logically equivalent to that given in class ($A' = \neg A$). However, on the assumption that we start in one of the legal states (for example, $((R \vee A) \oplus G)$), they are equivalent.

Check that: $((R \vee A) \oplus G) \rightarrow ((\neg A) \leftrightarrow (G \vee (R \wedge \neg A)))$ is a tautology.



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? **10**

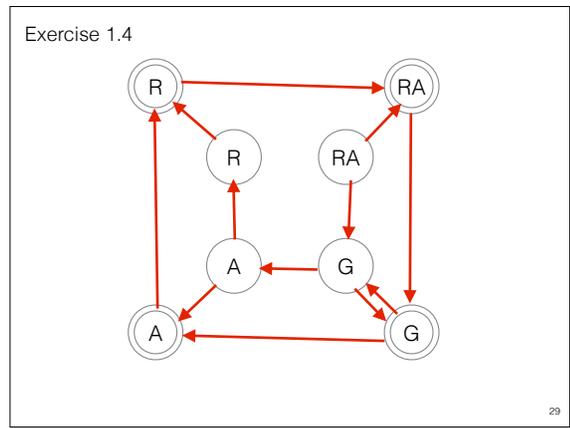
How many depend on only one variable? **4**

How many depend on no variables? **2**

Note the symmetries and duality apparent in the diagram.

If we think of each truth table as a set of valuations, then the sets are arranged by size (number of 1s in the truth table).

Of course we can use de Morgan, and expansions of \rightarrow and \oplus , etc., to give other equivalent expressions.



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.

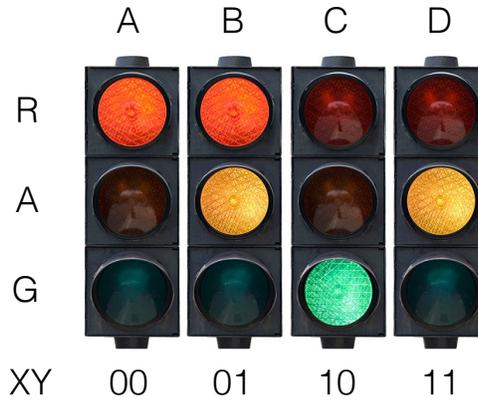
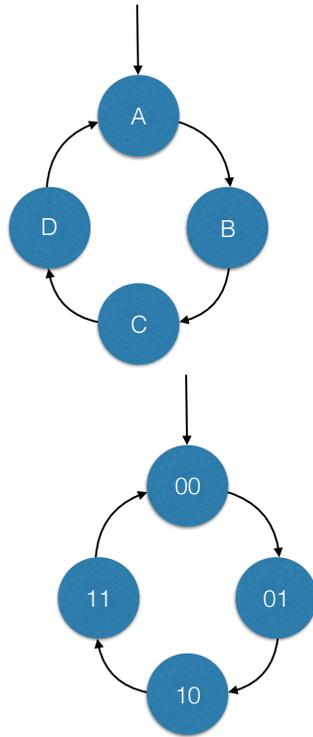
1. A car can only pass the light if it is green.
2. The light only changes from red to red-amber when a car is detected

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' ?

This machine responds to an external event, the arrival of a car. We shouldn't give an equation.

Discuss what may happen if a car breaks the rules, and runs a red light.



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

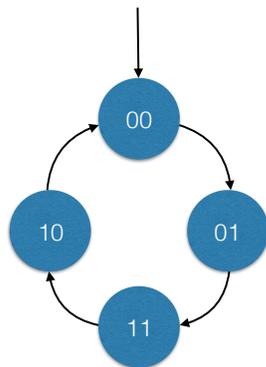
$$X' = X \oplus Y \text{ and } Y' = \neg Y$$

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.

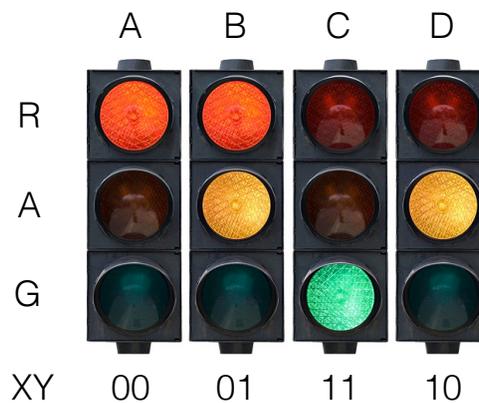


Give expressions for the next state logic

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

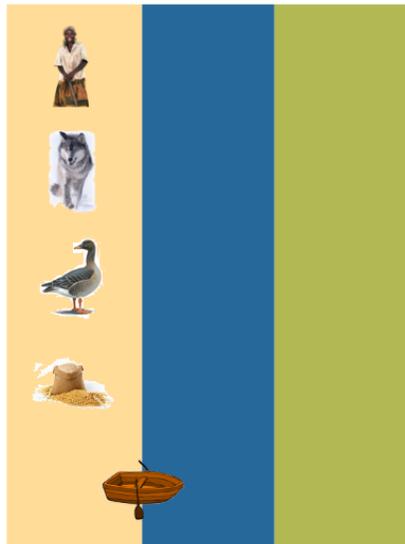
and the output logic

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





6.



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?

Here, we are seeking to characterise a set of possible locations of four entities, in which each can be²⁹ in exactly one of three places; to characterise a set of possible state transitions (an entity cannot get from one riverbank to the other without passing over the river); a set of illegal conditions; and from there work out which legal states are connected by possible transitions. The result is a logical model of the possible ways the farmer-goose-wolf-corn system could develop over time, which can then be used to characterise how the farmer can get from an initial state (everything west) to a goal state (everything east). If we accomplish this, is our description complete?

In fact, no. There are lots of other possible outcomes that could come about if an actual farmer, wolf, goose, and bag of corn. A wild animal might show up on one bank while the farmer is on the other and eat one of the unattended entities. The wolf might not be hungry. The goose might be large and aggressive. The model we have abstracts away from the actual situation, coding some regions of the actual range of possibilities into categories; sets that we indicate with the states of the model—thus WE encodes the (arbitrarily large) set of possible states of the world in which the wolf is on the east bank. Other possible states of the world (A robber or wild animal shows up, the boat springs a leak, a 50km asteroid annihilates all life on Earth) are simply not considered at all.

However, it should not be thought that this 'incompleteness'—the abstractions and idealisations involved in building a logical model of real-world conditions—is a weakness of the model. On the contrary, having an abstract model allows us to find solutions to problems that would be much harder to find in a more detailed model. It allows us to create a model that can be applied to multiple situations. It allows us to find patterns and explanations that of real phenomena that more detailed models would obscure.

NOTE OF CAUTION: There is another sense of the word 'complete', specific to formal systems such as logic and mathematics, which we will be introducing later in the course. Do not confuse the informal sense of the word used here with the technical sense that will come up later; they are quite different.

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	$(WW \oplus WB \oplus WE) \wedge \neg(WW \wedge WB \wedge WE)$	$\times 4$ (wolf,goose,corn,farmer)
not solo	$GB \rightarrow FB$	$\times 3$ (wolf,goose,corn)
no conflict	$GW \wedge (WW \vee CW) \rightarrow FW$	$\times 2$ (east, west)
no overload	$\neg(GB \wedge CB) \wedge \neg(GB \wedge WB) \wedge \neg(WB \wedge CB)$	$\times 1$

9

	West	East
	$\neg WW$	$\neg WE$
	$\neg CW$	$\neg CE$
	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 $\neg WW \leftrightarrow \neg WW$

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

$$WB \leftrightarrow (\neg WW \wedge \neg WE) \quad \times 4$$

$$WW \leftrightarrow \neg \neg WW \quad \times 8$$

This encoding uses only 8 propositional atoms – 256 states.

one place	$\neg WW \vee \neg WE$	$\times 4$
not solo	$(GW \wedge GE) \rightarrow (FW \wedge FE)$	$\times 3$
no conflict	$\neg GE \wedge (\neg WE \vee \neg CE) \rightarrow \neg FE$	$\times 2$
no overload	$(\neg GW \vee \neg GE \vee \neg CW \vee \neg CE) \wedge (\neg CW \vee \neg CE \vee \neg \neg WW \vee \neg \neg WE) \wedge (\neg \neg GW \vee \neg \neg GE \vee \neg \neg \neg WW \vee \neg \neg \neg WE)$	$\times 1$

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

- > Each instance of the nonsolo axiom is applicable to 18 states; however, since each set of the applicable states for each instance intersects with the set of each other instance for one third of it's states, and with both others for one ninth, with each application of an instance the total number of remaining states is reduced from 81 to 63, 51, and 43.
- > Each instance of the noconflict axiom reduces the remaining states by 8. These do not overlap, so the set of remaining states is reduced to 35, then 27.
- > The nooverload axiom reduces the set of remaining states by a further 7. This leaves a set of 20 legal states.

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

~~$\rightarrow XW \rightarrow XE$~~
 ~~$\rightarrow XE \rightarrow XW$~~

(Where X stands for one of the Farmer, the Goose, the Wolf, and the Corn)

These rules simply specify that nothing can move from one bank to the other in one time-step. This does not rule out moves to illegal states (such as everything getting in the boat at once), but it is not necessary to use the next-state rules to exclude possibilities already excluded by the legal-state axioms.

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	$A \rightarrow B$
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	
x		0	1
0	0	0	0
1	0	0	1

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

$x \rightarrow y$		y	
x		0	1
0	1	1	1
1	0	0	1

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

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

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