Informatics 1 - Computation & Logic: Tutorial 3

Counting

Week 5: 16-20 October 2016

Please attempt the entire worksheet in advance of the tutorial, and bring all work with you. Tutorials cannot function properly unless you study the material in advance. Attendance at tutorials is **obligatory**; please let the ITO know if you cannot attend.

You may work with others, indeed you should do so; but you must develop your own understanding; you can't phone a friend during the exam. If you do not master the coursework you are unlikely to pass the exams.

If we want to go beyond yes/no questions, it is natural to ask, *How many ...?* We are interested in sets, so we will ask how many elements there are in a set. We will focus on finite sets. We write $|A|$ or $\#A$ for the number of elements in A

I Let A and B be disjoint finite sets , with at least one element, $b \in B$ (A and B may have other elements).

(a) Is $b \in A$? No, by the definition of *disjoint set*.

Use arithmetic operators to give expressions for the following numbers:

(b) $|\{\}| =$ 0

(c) $|A \cup \{b\}| =$ $|A| + 1$
We know from I(a)a that $b \notin A$, so $A \cup \{b\}$ has one more member than A.

(d) $|\{\langle a, b\rangle \mid a \in A\}| =$ $|A|$

(e) $|A \cup B| =$ $|A| + |B|$
The general formula relating the size of sets to the size of their union is $|A| + |B| - |A \cap B|$: however, A and B are disjoint, which means that $|A \cap B| = \{\}$. But, as you should have found in I(b), $|\{\}| = 0$.[1]

(f) $|A \times B| =$ $|A| \times |B|$

(g) $|\wp A| =$ $2^{|A|}$
In selecting a subset of A, one has to make a binary decision for each member of A, determining

[1] When two sets are disjoint we may write $A + B$ for $A \cup B$.

whether the member will be in the subset. Thus each subset if the result of $|A|$ binary decisions, resulting in $2^{|A|}$ possible outcomes.

(h) $|\{f \mid f : A \to B\}| = \ |B|^{|A|}$
In selecting a mapping from each of the members of A to any of the members of B, for each member of A, one must decide between $|B|$ options; thus, with $|A|$ many $|B|$-fold decisions, there will be $|B|^{|A|}$ possible outcomes. Each subset of $X \subseteq A$ determines a function $x : A \longrightarrow \{0,1\}$, given by $x(a) = 1$ if $a \in X; x(a) = 0$ if $a \notin X$. The subset can be recovered from this function: $X = \{a \in A \mid x(a) = 1\}$. So, if we write 2 for $\{0,1\}$, your answer for I(g) should appear as a special case of the answer to this question.[2]

(i) $|\{R \subseteq A \times A \mid \text{R is a total ordering of } A\}| = \ |A|!$
An ordering of a set A is a binary relation R with some special properties (see Chapter 3 of MML).
Examples of orderings include the orderings, $<$, $\leq$, on natural numbers, or $\subset$, $\subseteq$ on the subsets of a set A

$$\leq \ = \{\langle x, y\rangle \in \mathbb{N} \times \mathbb{N} \mid x \leq y\} \qquad < \ = \{\langle x, y\rangle \in \mathbb{N} \times \mathbb{N} \mid x < y\}$$
$$\subseteq \ = \{\langle X, Y\rangle \in \wp A \times \wp A \mid X \subseteq Y\} \quad \subset \ = \{\langle X, Y\rangle \in \wp A \times \wp A \mid X \subset Y\}$$

Here we represent each relation as a set, so, for example,

$$x < y \text{ iff } \langle x, y\rangle \in < \qquad X \subseteq Y \text{ iff } \langle X, Y\rangle \in \subseteq$$

Notice that each ordering comes in two forms, weak $(\leq, \subseteq)$, and strong $(<, \subset)$. The weak and strong forms are closely related

$$x \leq y \text{ iff } x < y \text{ or } x = y \quad X \subset Y \text{ iff } X \subseteq Y \text{ and } X \neq Y \quad \text{(weak-strong)}$$

Each of these relations is *transitive*, for example,

$$x < y \text{ and } y < z \to x < z \quad X \subseteq Y \text{ and } Y \subseteq Z \to X \subseteq Z \quad \text{(transitive)}$$

The weak forms are *reflexive* and *antisymmetric*, while the strong forms are *irreflexive*. Examples:

$$x \leq y \text{ and } y \leq x \ \leftrightarrow \ x = y \qquad \text{(reflexive } (\leftarrow)\text{, antisymmetric } (\rightarrow))$$
$$X \subset Y \to X \neq Y \text{ (or equivalently, } X = Y \to X \not\subset Y) \qquad \text{(irreflexive)}$$

[2] One way to represent the natural numbers as sets is to define

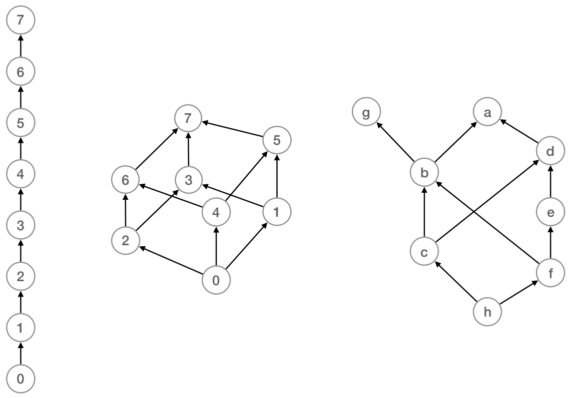
$$0 = \emptyset \qquad \text{(zero is emptyset)}$$
$$n + 1 = n \cup \{n\} \qquad \text{(the successor of } n \text{ has one extra element)}$$

Then $n = \{0, \ldots, n-1\}$ is a set with n elements, and $m < n$ iff $m \in n$.

These properties characterise weak and strong partial orders. A weak partial order, $\preceq$, is like $\subseteq$ and $\leq$ in that it is transitive, reflexive, antisymmetric relation. The related strong (partial) order, defined as in equations weak-strong, is transitive and irreflexive. We can also define a transitive, reflexive, antisymmetric relation $\preceq$ from a transitive irreflexive relation $\prec$, again using the *weak-strong* equations above. Thus each partial ordering can be represented by a strong or a weak ordering relation. Note that any transitive, irreflexive relation $x \prec y$ – that is, any weak partial order – is also *antisymmetric*, in the sense that $x \prec y \to y \not\prec x$.

Thus far we have been discussing the similarities between the orderings of subsets and of numbers. The key difference between the two examples is that when we draw a diagram of the two orderings, one is *linear* and the other, in general, is not.[3]

The diagram below shows three partial orders: the linear ordering of the numbers less than 8, the partial ordering of the eight subsets of $3 = \{0, 1, 2\}$, and an abstract partial ordering of eight elements $\{a, b, c, d, e, f, g, h\}$.



We say a weak partial ordering $\preceq$ of X is *total*, or *linear* iff

$$\textbf{for all } x, y \in X.\ x \preceq y \textbf{ or } y \preceq x\,.$$

The corresponding condition for totality of a strong ordering $\prec$ is that

$$\textbf{for all } x, y \in X.\ x \prec y \textbf{ or } y \prec x \textbf{ or } x = y\,.$$

The diagram shows some basic relationships. If there is an arrow $x \to y$ then $x < y$. But we also assume transitivity, so in the third diagram $f < g$ – since we have arrows $f \to b \to g$.

Each finite total order looks essentially like the first diagram. If A has n elements, then we can arrange them in increasing order, $x_0 \to x_1 \to \ldots \to x_{n-1}$, indexed by the numbers $0, \ldots, n-1$.

How many such arrangements are there? We have n choices for x_0, but

[3] When is the inclusion ordering on $\wp(X)$ linear?

then, since we can include each element only once, there are $n-1$ choices left for x_1, and so on – $n-i$ choices for x_i – until we have only one $(n-(n-1))$choice for x_{n-1}. So the total number of ways we can make these sucesive choices is $n \times (n-1) \times \ldots \times 1 = n!$.
This even works for the case $n = 0$. There is only one way to order the empty set, so $0! = 1$.

II Give rules, in the style of Tutorial 0, to generate the following sets:

(a) the set $\mathcal{F} \subseteq \wp\mathbb{N}$ of finite subsets of the natural numbers, $\mathbb{N}$.

$$\begin{aligned} \{\} &\in \mathcal{F} \\ n \in \mathbb{N}, F \in \mathcal{F} \to F \cup \{n\} &\in \mathcal{F} \end{aligned}$$

(b) the set $\mathbb{N}$ of natural numbers

$$\begin{aligned} 0 &\in \mathbb{N} \\ n \in \mathbb{N} \to n+1 &\in \mathbb{N} \end{aligned}$$

The natural numbers $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$ correspond to the sizes of finite sets. For a finite set, the answer to the question, *How many?*, will be a number. Since $\emptyset = \{\}$ is a finte set, 0 is a natural number.

In most living languages it is possible to name an arbitrary natural number. so, we can use natural language to give the answer. However, these names soon become unwieldy.

Tally marks are a unary numeral system. Each element of the set we are counting is represented by a separate mark, a *stroke*. For example, the numbers one, two, three are represented by |, ||, |||. To make this notation more easily legible, for larger numbers we use clusters. For example, four, five, six are represented by ||||, 𝍸, 𝍸|; twelve is represented by 𝍸𝍸||;

In our everyday lives, we usually use decimal notation for natural numbers. A finite sequence of n digits $x_i \in \{0, \ldots, 9\}$ represents a number.

$$\langle x_{n-1}, \ldots, x_0 \rangle \text{ represents } \sum_{i<n} 10^i x_i$$

Binary notation is similar. A finite sequence of n digits $x_i \in \{0, 1\}$ represents a number.

$$\langle x_{n-1}, \ldots, x_0 \rangle \text{ represents } \sum_{i<n} 2^i x_i$$

In general, for k-ary notation $(k > 1)$, a finite sequence of n digits $x_i \in \{0, \ldots, n-1\}$ represents a number.

$$\langle x_{n-1}, \ldots, x_0 \rangle \text{ represents } \sum_{i<n} n^i x_i$$

For n-ary notation with $n \leq 10$ we use the normal digits $0, \ldots. n-1$. We then move on to use letters of the alphabet as digits > 10. So the hexadecimal (16-ary) digits are $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$.

III Each row of the table below should show the same number represented in the various bases.

(a) Complete the table.

Base Name	2 binary	3 ternary	5	7	8 octal	10 decimal	16 hexadecimal
	1111	120	30	21	17	15	F
	1000	22	13	11	10	8	8
	10010	200	33	24	22	18	12
	10110	211	42	31	26	22	16
	101010110	110200	2332	666	526	342	156
	111000000	121121	3243	1210	700	448	1C0
	1010011010	220200	10131	1641	1232	666	29A
	10101011	20100	1141	333	253	171	AB

When we represent a number in base n, we use digits 0–$n-1$. Just as the places in decimal notation count units, tens, hundreds, thousands, etc., the places in n-ary notation represent units, ns, n^2s, n^3s, etc. Just as with decimal arithmetic, when we add multiply, subtract, or take powers of numbers in base n, the value in the units position of the result depends only on the value(s) in the units position of the argument(s).

The arithmetic of the units position is called arithmetic $\mod n$, (arithmetic modulo n). We write $x \bmod n$ for the value of the digit in the n-ary expansion of x. It is just the remainder of the integer division of x by n.

Both ($x \bmod n$), and the result, ($x \operatorname{div} n$), of the integer division, can be defined by the following properties:

$$0 \leq x \bmod n < n \quad x = n \times (x \operatorname{div} n) + (x \bmod n)$$

Note the connection between binary, octal and hexidecimal; three binary digits represent one octal digit; four binary digits (a nibble) represent one hex digit — and two nibbles form an eight-bit byte. One can divide any binary number into four-bit nibbles, in which each nibble is equivalent to one digit of hex; or divide it into 3-digit blocks equivalent to single digits in octal (adding leading zeros if required).
Consider the last four rows in the table above. Here is how the binary, octal, and hexidecimal line up.

Hex cols	256s				16s				1s			
Oct cols	512s			64s			8s			1s		
Bin cols	2048s	1024s	512s	256s	128s	64s	32s	16s	8s	4s	2s	1s
342 in hex	1				5				6			
...in oct	0			5			2			6		
...in bin	0	0	0	1	0	1	0	1	0	1	1	0
448 in hex	1				C				0			
...in oct	0			7			0			0		
...in bin	0	0	0	1	1	1	0	0	0	0	0	0
666 in hex	2				9				A			
...in oct	1			2			3			2		
...in bin	0	0	1	0	1	0	0	1	1	0	1	0
171 in hex	0				A				B			
...in oct	0			1			7			1		
...in bin	0	0	0	0	1	0	1	0	1	0	1	1

Thus we see that, for instance, where we get a 2 in octal, the corresponding binary triplet is 010; where we get an A in hexidecimal, the corresponding nibble is 1010: Oct 2 = Bin 010, Hex A = Dec 10 = Bin 1010.[4]

[4]You've come this far and have earned a joke. Why do computer scientists always mix up Hallowe'en and Christmas? Because 31(oct) = 25(dec).

(b) Complete the addition and multiplication tables for arithmetic mod $3, 5,$ and 7. Remember, this is just the arithmetic of the units column, so each square should contain just one digit in the range 0–$n-1$.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

IV This question concerns the 256 possible truth valuations of the following eight propositional letters A, B, C, D, E, F, G, H. For each of the following expressions, say how many of the 256 valuations satisfy the expression, and briefly explain your reasoning. For example, the expression D is satisfied by half of the valuations, that is 128 of the 256, since for each valuation that makes D true there is a matching valuation that make D false.

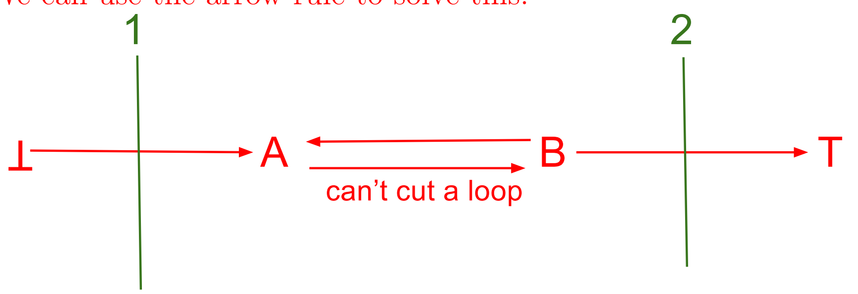
(a) $A \wedge B$ 64

(b) $(A \vee B) \wedge C$ 96

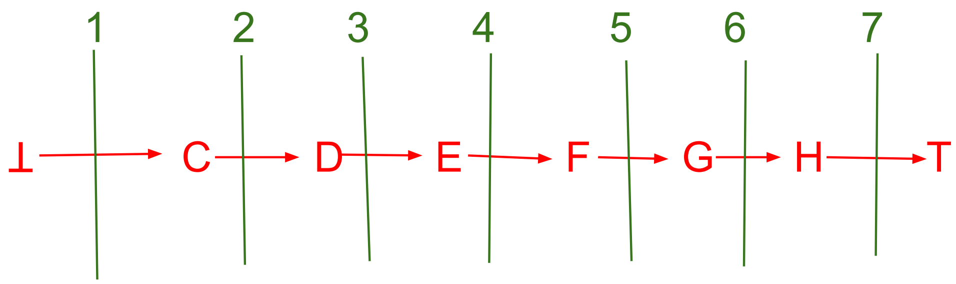
(c) $(A \to B) \to C$ 160

(d) $(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$

We can use the arrow rule to solve this:



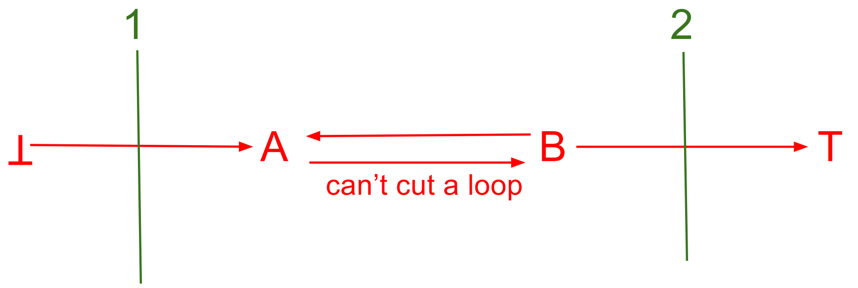
We find 2 valuations for A and B.



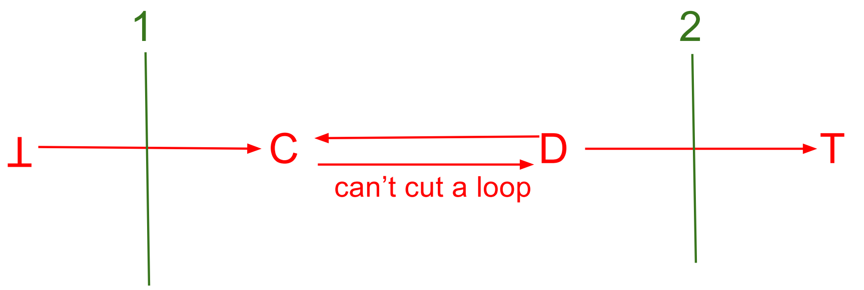
...and 7 for C, D, E, F, G, and H, giving $2 \times 7 = 14$ valuations in total

(e)

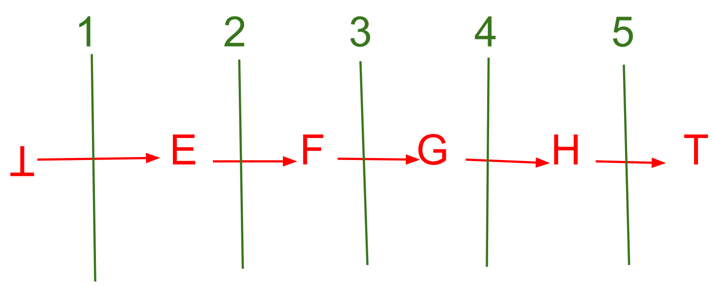
$$(A \to B) \wedge (B \to A) \wedge (C \to D) \wedge (D \to C) \\ \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$



We find 2 valuations for A and B.



...2 valuations for C and D.

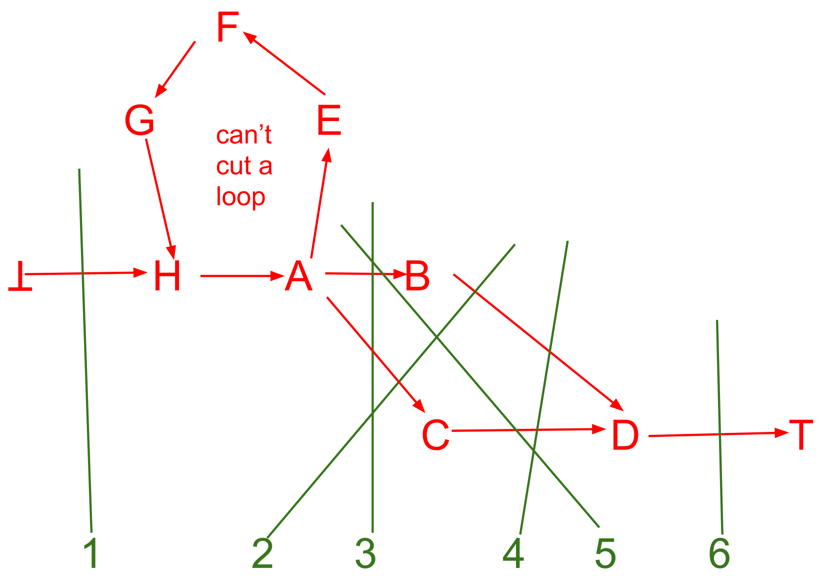


...and 5 for E, F, G, and H, giving $2 \times 2 \times 5 = 20$ valuations in total

(f)

$$(H \to A) \wedge (A \to B \wedge C) \wedge (B \vee C \to D) \wedge (A \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$

Noting that $A \to B \wedge C$ is equivalent to $(A \to B) \wedge (A \to C)$ and $(B \vee C \to D)$ is equivalent to $(B \to D) \wedge (C \to D)$, we derive the following graph, giving 6 satisfying valuations:



Tutorial Activities

1. As usual, buddy-up and take the first 20 minutes of the tutorial to check through your anwers to the homework exercises, I–IV.

   Ask others in your group, or call on one of the tutors if you have unresolved questions.

The main activity for this tutorial is on the next page. It introduces an idea that will be crucial to your understanding of the *resolution procedure* that is one of the key topics of this course.

Combining Constraints

In this exercise we consider a formula in conjunctive normal form (a conjunction of disjunctions of literals) as a set of constraints — each conjunction of literals is a constraint.

You should already have observed, while doing the tutorial exercises, that when we have two sets of constraints that are independent, in the sense that they share no common propositional letters, then we can solve each set of constraints separately, and then combine the answers.

2. Consider two sets of constraints

$$\Gamma = (R \vee B) \wedge (\neg A \vee G) \qquad\qquad \Delta = (\neg R \vee A) \wedge (\neg B \vee G)$$

(a) How many of the sixteen states of R, B, A, G satisfy Γ?
9 – there are 3 ways of satisfying each constraint.

(b) How many satisfy Δ? 9

(c) Use the distributive law to write down the CNF for $\Gamma \vee \Delta$. This gives a set of constraints that is satisfied by exactly those states that satisfy either Γ or Δ or both.

Hints: In algebra $(ab + cd)(wx + yz) = abwx + abyz + cdwx + cdyz$.
In logic any constraint that includes both an atom and its negation is trivially satisfied, and can be omitted.

$$\Gamma \vee \Delta = ((R \vee B) \wedge (\neg A \vee G)) \vee ((\neg R \vee A) \wedge (\neg B \vee G)) =$$
$$(R \vee B \vee \neg R \vee A) \wedge (\neg A \vee G \vee \neg R \vee A) \wedge (R \vee B \vee \neg B \vee G) \wedge (\neg A \vee G \vee \neg B \vee G)$$
$$= \neg A \vee G \vee \neg B$$

(d) How many states of $RBAG$ satisfy $\Gamma \vee \Delta$?
14 – all except the two states that satisfy $A \wedge \neg G \wedge B$.

(e) How many states satisfy $\Gamma \wedge \Delta$?
4 = 9+ 9 - 14
To understand this calculation consider: The set of states satisfying Γ consists of two disjoint subsets: those satisfying $\Gamma \wedge \Delta$ and those satisfying $\Gamma \wedge \neg\Delta$. The set satisfying Δ also consists two disjoint subsets: those satisfying $\Gamma \wedge \Delta$ and those satisfying $\neg\Gamma \wedge \Delta$. The set satisfying $\Gamma \vee \Delta$ consists of three: $\Gamma \wedge \Delta$, $\Gamma \wedge \neg\Delta$, and $\neg\Gamma \wedge \Delta$. Remember from q1.e of the homework that the cardinality of the union of disjoint sets is always equal to the sum of the cardinalities of the sets individually. Thus,

$$|\Gamma| + |\Delta| = 2|\Gamma \wedge \Delta| + |\Gamma \wedge \neg\Delta| + |\neg\Gamma \wedge \Delta|$$

and

$$|\Gamma \vee \Delta| = |\Gamma \wedge \Delta| + |\Gamma \wedge \neg\Delta| + |\neg\Gamma \wedge \Delta|$$

therefore

$$|\Gamma| + |\Delta| - |\Gamma \vee \Delta| = |\Gamma \wedge \Delta|$$

3. Consider the following set of constraints:

$$\Omega = (X \vee R \vee B) \wedge (X \vee \neg A \vee G) \wedge (\neg X \vee \neg R \vee A) \wedge (\neg X \vee \neg B \vee G)$$

   How many states of the five boolean variables $XRBAG$ satisfy Ω?

   Hint: Divide the states that satisfy Ω into two disjoint subsets by considering separately the states where X is true and the states where $\neg X$ is true, then refer to the previous question.

18 – the states where X is true must satisfy Δ; those where X is false must satisfy Γ. These two sets are disjoint (they have no elements in common, because the elements of one have $X = \top$ and the elements of the other have $X = \bot$).

This tutorial exercise sheet was written by Michael Fourman and Dave Cochran. Send comments to `Michael.Fourman@ed.ac.uk`