Informatics 1 - Computation & Logic: Tutorial 3

Counting

Week 5: 16-20 October 2017

> 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 $b \in A$?

Use arithmetic operators to give expressions for the following numbers:

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

(c) $|A \cup \{b\}| =$

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

(e) $|A \cup B| =$

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

(g) $|\wp A| =$

(h) $|\{f \mid f : A \to B\}| =$

(i) $|\{R \subseteq A \times A \mid \text{R is a total ordering of } A\}| =$

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, $\mathbb{N}$.

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

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						
		200					
			42				
				666			
					700		
						666	
							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 div n), of the integer division, can be defined by the following properties:

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

(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			
1			
2			

$\times$	0	1	2
0			
1			
2			

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

$\times$	0	1	2	3	4
0					
1					
2					
3					
4					

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

$\times$	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							

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) $A \wedge B$

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

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

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

(e)

$$
\begin{aligned}
(A \to B) \land (B \to A) \land (C \to D) \land (D \to C)& \\
\land (E \to F) \land (F \to G) \land (G \to H)&
\end{aligned}
$$

(f)

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

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 ?

   (b) How many satisfy Δ ?

   (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.

   (d) How many states of $RBAG$ satisfy $\Gamma \vee \Delta$?

   (e) How many states satisfy $\Gamma \wedge \Delta$?

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.

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