DISCRETE MATH 37181 HOMEWORK SHEET 7

INSTRUCTIONS. Try these sometime after your tutorial and before the next lecture. Set aside some time each week to keep up with the homework. Partial solutions at the end of the PDF.

1. Write the base-10 number 341 (three hundred and forty one) as $\sum b_i 2^i$ for $b_i \in \{0,1\}$. [1]

2. (a) State the binomial theorem, and draw Pascal's triangle down to level 6.

   (b) Expand $(a+b)^6$.

3. Define a relation $\mathscr{R}$ on the set of all finite length binary strings by $b_1 \mathscr{R} b_2$ if

   – the sum of the digits in b_1 is strictly greater that the sum of the digits in b_2, or
   – b_2 is obtained from b_1 by swapping two digits in distinct positions.

   For each option below, state whether it is true or false. Explain each answer.

   A. $\mathscr{R}$ is reflexive

   B. $\mathscr{R}$ is antisymmetric

   C. $\mathscr{R}$ is transitive

   D. $0110 \mathscr{R} 1110$

   E. $\mathscr{R}$ is symmetric

   F. $0111 \mathscr{R} 1110$

   G. $\mathscr{R}$ is a partial order

   H. $\mathscr{R}$ is an equivalence relation

4. (Note this is an advanced, but cool, question. (b), (c) will not be on the exam or LPC.)

   Recall from the Tutorial Worksheet that a *boolean function* is a function of the form $f : \{0,1\}^n \to \{0,1\}$.

   A *boolean circuit* in n variables $x_1, \ldots, x_n$ of size ℓ is a sequence $(s_1, \ldots, s_\ell)$, where each s_i is in one of the following forms:
   – $\text{op}_1 \vee \text{op}_2$, where op_1 (resp. op_2) is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0,1\}$.
   This $\vee$ represents the boolean *or* operation, so $\vee : \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined as $\vee(x,y) = 0$ if and only if $x = y = 0$.
   – $\text{op}_1 \wedge \text{op}_2$, where op_1 (resp. op_2) is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0,1\}$.
   This $\wedge$ represents the boolean *and* operation, so $\wedge : \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined as $\wedge(x,y) = 1$ if and only if $x = y = 1$.

Date: Week 7.

[1] i.e. write 341 in binary, or as its 2-ary expansion. For example, $35 = 32+2+1 = 1.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 1.2^0$ so in binary $32 = 100011_2$

– $\neg$op, where op is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0, 1\}$.
This $\neg$ represents the boolean *not* operation, so $\neg : \{0,1\} \to \{0,1\}$ is defined as $\neg(x) = 1$ if and only if $x = 0$.

(a) What is the number of boolean functions of the form $f : \{0,1\}^n \to \{0,1\}$? [2]

(b) What is the number of boolean circuits in n variables of size ℓ? If you wish, you can just give an estimation of the upper bound.

(c) Conclude that when $n \geqslant 10$, there exists a boolean function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by any boolean circuit in n variables of size n^2.
[3]

[2] already answered on Tutorial Worksheet 7

[3] An extension of this question shows that polynomial-size boolean circuits cannot compute all boolean functions. This observation goes back to Claude Shannon, the founder of information theory.

Brief solutions:

1. $341 = 256 + 64 + 16 + 4 + 1$

2. (b) $a^6 + 6a^5b + 15a^4b^2 + 20a^3b^3 + 15a^2b^4 + 6ab^5 + b^6$

3. **A**. 01 is not related to 01, so not reflexive.

   B. $01 \mathscr{R} 10$ and $10 \mathscr{R} 01$ (swapping digits) so not antisymmetric.

   C. $0101 \mathscr{R} 1100$ and $1100 \mathscr{R} 0111$ but 0101 is not related to 0111 since the sum of digits in 0101 is 2 and in 0111 is 3 (so not swapped digits, and fails digit sum condition) therefore it is not transitive.

   D. $0110 \mathscr{R} 1110$ is false since the sum of digits in the first string is smaller, and you cannot swap digits to go between.

   E. $01 \mathscr{R} 00$ but 00 is not related to 01, so not symmetric.

   F. $0111 \mathscr{R} 1110$ is true since you can swap the first and last digits

   G. Since its not reflexive, it is not a partial order.

   H. Since its not reflexive, it is not an equivalence relation.

4. (a) 2^{2^n}. (See Worksheet 7)

   (b) For each s_i, there are three types for the operation (one of $\{\vee, \wedge, \neg\}$), and $n+i+1$ possibilities for each operand. So there are at most $2(n+i+1)^2 + (n+i+1) = (n+i+1)(2n+2i+3)$ possibilities for s_i. The number of length-ℓ boolean circuits is therefore

   $$\prod_{i=1}^{\ell} \big((n+i+1)(2n+2i+3)\big).$$

   A convenient upper bound is then

   $$(n+\ell+1)^{\ell}(2n+2\ell+3)^{\ell} \leqslant (2n+2\ell+3)^{2\ell}.$$

(c) When $\ell = n^2$, then the above upper bound becomes $2^{2n^2 \cdot \log_2(2n+2n^2+3)}$. Recall that the number of boolean functions is 2^{2^n}. So when n is large enough, $2n^2 \cdot \log_2(2n + 2n^2 + 3)$ is less than 2^n, so the desired conclusion follows. (You may work out how large n needs to be for $2n^2 \cdot \log_2(2n + 2n^2 + 3) < 2^n$, but that's probably not the most interesting part.)

As noted in the footnote, this question shows that polynomial-size boolean circuits cannot compute all boolean functions. This observation goes back to Claude Shannon, the founder of information theory. It may be viewed as a finite version of Cantor's diagonalisation method (which we were using when we showed $\mathbb{R}$ is not countable) which later was used by Alan Turing to show that there exist undecidable algorithmic problems for Turing machines.

Thanks to Youming Qiao for this question.