Fall 2022
© Daniel Leivant

Symbols and Sets

Aug 2022

Set operations

1. Show that $\cup$ distributes over $\cap$, that is: for all sets $A, B, C,$

   Solution. $$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

   $$\begin{array}{lll} x \in A \cup (B \cap C & \text{IFF} & \text{either} \quad x \in A \quad \text{or} \quad x \in B \cap C \\ & \text{IFF} & \text{either} x \in A \quad \text{or both} \quad x \in B \quad \text{and} \quad x \in C \\ & \text{IFF} & x \in A \quad \text{or} \quad x \in B \quad \text{as well as} \quad x \in A \quad \text{or} \quad x \in C \\ & \text{IFF} & x \in A \cup B \quad \text{and} \quad x \in A \cup C \\ & \text{IFF} & x \in (A \cup B) \cap (A \cap C) \end{array}$$

2. Give elements of $\mathcal{P}(\mathbb{N})$ that are:

   (a) Finite non-empty (fewer than 5 elements).

   Solution. $\{1\}$

   (b) Co-finite (i.e. have a finite complement).

   Solution. $\mathbb{N}$, $\mathbb{Z}^+$.

   (c) Neither finite nor co-finite.

   Solution. The set of even natural numbers.

3. Call a subset of $\mathbb{N}$ basic if it is defined from finite sets by using (repeatedly, if needed) the basic set operations of difference, union and intersection.
   Show that a set $A \subseteq \mathbb{N}$ is basic iff it is finite or co-finite. [The proof is by induction, and will be given in the section on Inductive Sets.]

Symbolic versus numeric computing

(a) The grade-school addition algorithm is an example of a *symbolic* algorithm: it works for positional numerals (such as decimal and binary) but not for others (such as Roman numerals).

In contrast, the following program-outline is a *numeric* algorithm for division, because the representation is immaterial (addition is assumed given, and is not part of the algorithm).

$q := 0;\ \ s := 0;$
while $\ s < n\ $ **repeat**
$\quad s := s + d;\ \ q := q + 1$
end

Upon termination q contains the quotient n/d rounded up.

For each of the following algorithms, determine whether it is symbolic or numeric, and briefly explain your answer.

i. Long division.

Solution. A symbolic algorithm, good only for positional numerals. For example, it does not work for unary numerals or for Roman numerals.

ii. High-school square root (see e.g. `https://www.basic-mathematics.com/square-root-algorithm.html`)

Solution. Same answer as (a).

iii. Finding $\log_2 n$ ($n > 0$) by repeatedly multiplying by 2 (starting from 1) until n is exceeded (multiplication is given).

Solution. A numeric algorithm. Number representation is not at stake here.

iv. Finding whether a number is a multiple of 10 by examining the last digit.

Solution. Symbolic algorithm, specific to decimal numerals, that does not even work for other positional numerals.

v. Euclid's algorithm for the greatest common divisor of positive integers a and b (taking subtraction as given):

$x := a;\ y := b$
while $x \neq y$ **repeat**
  if $x < y$ **then** $y := y - x$
  else $x := x - y$
end

(The final value of x is the greatest common divisor of the inputs a, b.)
Solution. Numeric algorithm. Representation is not at stake.

(b) Consider the following numeral system:

Numerals are strings $w \in D$ where $D = \{1, 2\}^*$. Their successive numeric values: $[\mathtt{1}] = 1$, $[\mathtt{2}] = 2$, $[\mathtt{11}] = 3$, $[\mathtt{12}] = 4$, $[\mathtt{21}] = 5$, $[\mathtt{22}] = 6$, $[\mathtt{111}] = 7$, etc.

i. What is the numeric value $[w]$ of a string $w = d_k \cdots d_2 d_1 d_0$? (You may use addition, multiplication, and exponentiation.)

Solution. $[d_k \cdots d_2 d_1 d_0] = \sum_{i=0}^{k} = d_k \cdot 2^k + \cdots + d_1 \cdot 2 + d_0$

ii. Describe in words an algorithm that computes the successor function for the numeral notation above. That is, your algorithm should convert a $w \in \{\mathtt{1}, \mathtt{2}\}^*$ into the string representing $[w] + 1$. For example, `22` (the numeral for 6) is converted to `111` (the numeral for 7).
For comparison, here's the usual algorithm for converting a *binary* numeral for n into the binary numeral for $n+1$.
Starting from the rightmost digit, repeat:

- *If the digit is* `0` *replace it by* `1`*, and stop.*
- *If the digit is* `1`*, replace it by* `0` *and step left.*
- *If you've passed the leftmost digit, place a* `1`*, and stop.*

Self-reference and the Separation Principle

(a) Prove the following statements. You may use the Separation Principle in your proofs.

i. There is no such thing as the "set of all sets."

Solution. If U were the set of all sets, then Russel's paradoxical definition

$$R = \{x \in U \mid x \notin x\}$$

would be a legitimate set, by the Separation Principle. And that leads to a contradiction, as we know.

ii. There is no such thing as the "set of all singletons".

Solution. If a set S of all singletons exists, by Separation we'd get the set

$$Q = \{\{x\} \in S \mid \{x\} \notin x\}$$

For every singleton set $\{x\}$ we'd have $\{x\} \in Q$ iff $\{x\} \notin x$. Since Q is a legitimate set, we can take x to be Q, and get that $\{Q\} \in Q$ iff $\{Q\} \notin Q$, a contradiction.