HW1 Definitions and Notation

CSE20F21

Sample Solutions

Assigned questions

1. (*Graded for correctness*[1]) Each of the sets below is described using set builder notation or as a result of set operations applied to other known sets. Rewrite them using the roster method.

   Remember our discussions of data-types: use clear notation that is consistent with our class notes and definitions to communicate the data-types of the elements in each set.

   Sample response that can be used as reference for the detail expected in your answer:

   The set $\{\texttt{A}\} \circ \{\texttt{AU}, \texttt{AC}, \texttt{AG}\}$ can be written using the roster method as

   $$\{\texttt{AAU}, \texttt{AAC}, \texttt{AAG}\}$$

   because set-wise concatenation gives a set whose elements are all possible results of concatenating an element of the left set with an element of the right set. Since the left set in this example only has one element ($\texttt{A}$), each of the elements of the set we described starts with $\texttt{A}$. There are three elements of this set, one for each of the distinct elements of the right set.

   (a) $$\{x \in S \mid rnalen(x) = 1\} \circ \{x \in S \mid rnalen(x) = 1\}$$

   where S is the set of RNA strands and *rnalen* is the recursively defined function that we discussed in class.

[1]This means your solution will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

Solution: This set can be written using the roster method as

$$\{\texttt{AA}, \texttt{AC}, \texttt{AU}, \texttt{AG}, \texttt{CA}, \texttt{CC}, \texttt{CU}, \texttt{CG}, \texttt{GA}, \texttt{GC}, \texttt{GU}, \texttt{GG}, \texttt{UA}, \texttt{UC}, \texttt{UU}, \texttt{UG}\}$$

because (as in the example) we need to find all possible results of concatenating an element of the left set with an element of the right set. For this set, we notice that both sets in the set-wise concatenation are the same and are defined (using set builder notation) as the collection of all RNA strands for which the function *rnalen* gives the value 1. Looking at the definition of *rnalen* from class, we see that it gives the value 1 in the basis step and any function application that applies the recursive step will give a higher value. Thus, the set that is being set-wise concatenated with itself is the collection of all RNA strands mentioned in the basis step of *rnalen*, namely $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{U}\}$. The set we specfied above with roster method is the result of concatenating two (possibly equal) choices from this set.

(b)

$$\{(r, g, b) \in C \mid r + g + b = 1\}$$

where $C = \{(r, g, b) \mid 0 \leq r \leq 255, 0 \leq g \leq 255, 0 \leq b \leq 255, r \in \mathbb{N}, g \in \mathbb{N}, b \in \mathbb{N}\}$ is the set that you worked with in Monday's review quiz.

Solution: This set can be written using the roster method as

$$\{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$$

because the only allowed values for the r, g, b components in the 3-tuple are nonnegative integers, which means that their smallest positive value is 1. Thus, if two or more of the components in a 3-tuple are nonzero, the sum of the components will be greater than 1, which is not allowed for 3-tuples that satisfy the equation specified in the set-builder definition of the set. Thus, the elements of the set are obtained by all possible ways of choosing the one component that is non-zero (and has value 1 to make the equation specified in the set-builder notation true).

(c)

$$\{a \in \mathbb{Z} \mid a \textbf{ div } 2 = a \textbf{ mod } 2\}$$

Solution: This set can be written using the roster method as

$$\{0, 3\}$$

. We see from the definition of **mod** that when we calculate a **mod** 2 for an integer a, there are only two possible remainders: 0 (when a is even) and 1 (when a is odd). We consider each possibility in turn: if $a \textbf{ mod } 2 = 0$ and we want $a \textbf{ div } 2 = a \textbf{ mod } 2 = 0$ then, by the definition of **div** and **mod** in the Division Algorithm, $a = 0 \cdot 2 + 0 = 0$. Similarly, if $a \textbf{ mod } 2 = 1$ and we want $a \textbf{ div } 2 = a \textbf{ mod } 2 = 1$ then, by the definition of **div** and **mod** in the Division Algorithm, $a = 1 \cdot 2 + 1 = 3$.

2. (*Graded for fair effort completeness*[2])

[2]This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers.

(a) In Wednesday's review quiz, you considered some attempted recursive definitions for the function with domain $\mathbb{N}$ and with codomain $\mathbb{Z}$ which gives 2^n for each n. Write out a correct recursive definition of this function.

Solution:

Basis Step: $2^0 = 1$

Recursive Step: If $n \in \mathbb{N}$, then $2^{n+1} = 2 \cdot 2^n$

Informally, this recursive definition matches the rule for computing powers of 2 because to get to the "next" power of 2, we multiple the "previous" one by 2.

(b) How would your answer to part (a) change if we consider a new function with the same domain and rule but whose codomain is $\mathbb{R}$?

Solution: The answer wouldn't change because the the recursive definition tells us how to apply the rule (which is the same) to every element in the domain (which is the same). The codomain tells us what the type of outputs is. In the new function, we treat the powers of 2 as real numbers, which is okay because every integer (whole number) is a real number (since it is on the number line).

(c) How would your answer to part (a) change if we consider a new function with the same codomain and rule but whose domain is $\mathbb{R}$?

Solution: We would not be able to write a recursive rule for this new function because its domain is not given recursively so the answer would need to completely change.

(d) Write a recursive definition of the function with domain $\mathbb{Z}^+$, codomain $\mathbb{Z}^+$ and which gives $n!$ for each n. The ! symbol is the "factorial" symbol and means that we need to multiple n by each of the integers between it and 1 inclusive. For example, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$.

Solution: A recursive definition for $\mathbb{Z}^+$ has Basis step: $1 \in \mathbb{Z}^+$ and Recursive step: If $n \in \mathbb{Z}^+$ then $n + 1 \in \mathbb{Z}^+$. Mirroring this recursive structure, we give a recursive definition for the rule of the factorial function:

Basis Step: $1! = 1$

Recursive Step: If $n \in \mathbb{N}$, then $(n + 1)! = (n + 1)n!$

Informally, this recursive definition matches the rule for computing factorial because each "next" factorial application has a new term being multiplied with all the previous ones.

Notice that this recursive definition is for a **function**, not a **set**, so the verb in each of the steps is $=$ (not $\in$). In other words, each step tells us the result of applying the function to an input from the domain, and the domain is a recursively defined set so the inputs can build on one another.

3. (*Graded for correctness*) Recall the function d_0 which takes an ordered pair of ratings 3-tuples and returns a measure of the distance between them given by

$$d_0(\ (\ (x_1, x_2, x_3), (y_1, y_2, y_3)\ )\ ) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

Sample response that can be used as reference for the detail expected in your answers for this question:

To give an example of two 3-tuples that are d_0 distance 1 from each other, consider the 3-tuples $(1, 0, 0)$ and $(0, 0, 0)$. We calculate the function application:

$$d_0(\ (\ (1, 0, 0), (0, 0, 0)\ )\ ) = \sqrt{(1-0)^2 + (0-0)^2 + (0-0)^2} = \sqrt{1^2 + 0^2 + 0^2} = \sqrt{1} = 1,$$

which is the result required for this example.

(a) Give an example of three 3-tuples

$$\begin{gathered}(x_{1,1}, x_{1,2}, x_{1,3}) \\ (x_{2,1}, x_{2,2}, x_{2,3}) \\ (x_{3,1}, x_{3,2}, x_{3,3})\end{gathered}$$

that are all d_0 distance **greater than** 1 from each other. In other words, for each i and j between 1 and 3 (with $i \neq j$),

$$d_0(\ (\ (x_{i,1}, x_{i,2}, x_{i,3}), (x_{j,1}, x_{j,2}, x_{j,3})\ )\ ) > 1$$

Your answer should include **both** specific values for each example 3-tuple **and** a justification of your examples with (clear, correct, complete) calculations and/or references to definitions and connecting them with the desired conclusion.

To think about: how will you justify that the square root of a number is greater than 1? Are calculations from a calculator accurate enough to help us?

Solution: To give an example of three 3-tuples that are d_0 distance greater than 1 from each other, consider the 3-tuples $(-1, -1, -1)$ and $(0, 0, 0)$ and $(1, 1, 1)$, and calculate the d_0 distance between each of the three pairs of these. Note that we only need to consider three pairs because order doesn't matter in calculating d_0. We calculate the function application:

$$\begin{aligned}d_0(\ (\ (1, 1, 1), (-1, -1, -1)\ )\ ) &= \sqrt{(1 - -1)^2 + (1 - -1)^2 + (1 - -1)^2} = \sqrt{2^2 + 2^2 + 2^2} \\ &= \sqrt{12} > \sqrt{9} = 3 > 1,\end{aligned}$$

as required for this example. Notice that we used the fact that square root is an increasing function. Next, we calculate the function application:

$$\begin{aligned}d_0(\ (\ (1, 1, 1), (0, 0, 0)\ )\ ) &= \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \sqrt{1^2 + 1^2 + 1^2} \\ &= \sqrt{3} > \sqrt{1} = 1\end{aligned}$$

as required for this example. We again used the fact that square root is an increasing function. Last, we calculate the function application:

$$\begin{aligned} d_0(\ (\ (-1,-1,-1),(0,0,0)\ )\ ) &= \sqrt{(-1-0)^2+(-1-0)^2+(-1-0)^2} \\ &= \sqrt{(-1)^2+(-1)^2+(-1)^2} = \sqrt{3} > \sqrt{1} = 1 \end{aligned}$$

like before. Thus, this example of three 3-tuples works.

(b) What is the range of values that results from applying the function d_0 to ordered pairs of 3-tuple ratings? That is, what are the smallest and largest possible results?

Your answer should include **both** specific values for the smallest and largest possible results **and** a justification of your answers with (clear, correct, complete) calculations and/or references to definitions and connecting them with the desired conclusion.

Solution: The smallest value resulting from a function application would be when each of the terms being summed together inside the square root is 0 (because they have to be nonnegatives since they're the result of squaring some quantity). This means that the difference between each component of the tuples must be 0, or in other words that the ordered pair of ratings which is input to d_0 has equal ratings. In this case, we have $d_0(\ (\ (x_1,x_2,x_3),(x_1,x_2,x_3)\ )\ )$ for some x_1,x_2,x_3 each in $\{-1,0,1\}$ and the function application gives

$$\sqrt{(x_1-x_1)^2+(x_2-x_2)^2+(x_3-x_3)^2} = \sqrt{0+0+0} = 0.$$

Thus, the smallest value that results from applying the function d_0 to ordered pairs of 3-tuple ratings is 0.

The largest possible result will be when each of the terms in the square root is as big as possible, which means that the ratings for that term are are different as possible: 1 and -1 (in some order). In this case, this term in the sum would be $(1--1)^2$ or $(-1-1)^2$, which (in either case) gives 4. When each term has this maximum value, the function application gives

$$\sqrt{4+4+4} = \sqrt{12} = 2\sqrt{3}$$

and results from ordered pairs such as $((1,-1,-1),(-1,1,1))$ or $((1,1,1),(-1,-1,-1))$, etc. Thus, the largest value that results from applying the function d_0 to ordered pairs of 3-tuple ratings is $2\sqrt{3}$.