15 Recursively Defined Sets

Two different ways of defining a set have been discussed. We can describe a set by the roster method, listing all the elements that are to be members of the set, or we can describe a set using set-builder notation by giving a predicate that the elements of the set are to satisfy. Here we consider defining sets in another natural way: recursion.

15.1 Recursive definitions of sets

Recursive definitions can also be used to build sets of objects. The spirit is the same as for recursively defined sequences: give some initial conditions and a rule for building new objects from ones already known.

Example 15.1. *For instance, here is a way to recursively define the set of positive even integers, E. First the initial condition:* $2 \in E$*. Next the recursive portion of the definition: If* $x \in E$*, then* $x + 2 \in E$*. Here is what we can deduce using these two rules. First of course, we see* $2 \in E$ *since that is the given initial condition. Next, since we know* $2 \in E$*, the recursive portion of the definition, with x being played by* 2*, says* $2 + 2 \in E$*, so that now we know* $4 \in E$*. Since* $4 \in E$*, the recursive portion of the definition, with x now being played by* 4*, says* $4 + 2 \in E$*, so that now we know* $6 \in E$*. Continuing in this way, it gets easy to believe that E really is the set of positive even integers.*

Actually, there is a little more to do with example 15.1. The claim is that E consists of exactly all the positive even integers. In other words, we also need to make sure that no other things appear in E besides the positive even integers. Could 312211 somehow have slithered into the set E? To verify that such a thing does not happen, we need one more fact about recursively defined sets. The only elements that appear in a set defined recursively are those that make it on the basis of either the initial condition or the recursive portion of the definition. No elements of the set appear, as if by magic, from nowhere.

In this case, it is easy to see that no odd integers sneak into the set. For if so, there would be a smallest odd integer in the set and the only way it could be elected to the set is if the integer two less than it were in the set. But that would mean a yet smaller odd integer would be in the set, a contradiction. We won't go into that sort of detail for the following examples in general. We'll just consider the topic at the intuitive level only.

Example 15.2. *Give a recursive definition of the set, S, of all nonnegative integer powers of* 2*.*

Initial condition: $1 \in S$*. Recursive rule: If* $x \in S$*, then* $2x \in S$*. Applying the initial condition and then the recursive rule repeatedly gives the elements:*

$$1 \qquad 2 \cdot 1 = 2 \qquad 2 \cdot 2 = 4 \qquad 2 \cdot 4 = 8 \qquad 2 \cdot 8 = 16$$

and so on, and that looks like the set of nonnegative powers of 2*.*

Example 15.3. *A set, S, is defined recursively by*

(1) (initial conditions) $1 \in S$ *and* $2 \in S$*, and*

(2) (recursive rule) If $x \in S$*, then* $x + 3 \in S$*. Describe the integers in* S*.*

The plan is to use the initial conditions and the recursive rule to build elements of S until we can guess a description of the integers in S. From the initial conditions we know $1 \in S$ *and* $2 \in S$*. Applying the recursive rule to each of those we get* $4, 5 \in S$*, and using the recursive rule on those gives* $7, 8 \in S$*, and so on.*

So we get $S = \{1, 2, 4, 5, 7, 8, 10, 11, \cdots\}$ *and it's apparent that S consists of of the positive integers that are not multiples of* 3*.*

15.2 Sets of strings

Recursively defined sets appear in certain computer science courses where they are used to describe sets of strings. To form a string, we begin with an **alphabet** which is a set of symbols, traditionally denoted by Σ. For example $\Sigma = \{a, b, c\}$ is an alphabet of three symbols, and $\Sigma = \{!, @, \#, \$, \%, \&, X, 5\}$ is an alphabet of eight symbols. A **string** over the alphabet Σ is any finite sequence of symbols from the alphabet. For example *aaba* is a string of **length** four over the alphabet $\Sigma = \{a, b, c\}$, and !!5$X$$$5@@ is a length nine string over $\Sigma = \{!, @, \#, \$, \%, \&, X, 5\}$. There is a special string over any alphabet denoted by λ called the **empty string**. It contains no symbols, and has length 0.

Example 15.4. *A set, S, of strings over the alphabet $\Sigma = \{a, b\}$ is given recursively by (1) $\lambda \in S$, and (2) If $x \in S$, then $axb \in S$. Describe the strings in S.*

The notation axb means write down the string a followed by the string x followed by the string b. So if $x = aaba$ then $axb = aaabab$. Let's experiment with the recursive rule a bit, and then guess a description for the strings in S. Starting with the initial condition we see $\lambda \in S$. Applying the recursive rule to λ gives $a\lambda b = ab \in S$. Applying the recursive rule to ab gives $aabb \in S$, and applying the recursive rule to aabb shows $aaabbb \in S$. It's easy to guess the nature of the strings in S: Any finite string of a's followed by the same number of b's.

Example 15.5. *Give a recursive definition of the set S of strings over $\Sigma = \{a, b, c\}$ which do not contain adjacent a's. For example ccabbbabba is acceptable, but abcbaabaca is not.*

For the initial conditions we will use (1) $\lambda \in S$, and $a \in S$. If we have a string with no adjacent a's, we can extend it by adding b or c to either end. But we'll need to be careful when adding more a's. For the recursive rule we will use (2) if $x \in S$, then $bx, xb, cx, xc \in S$ and $abx, xba, acx, xca \in S$.

Notice how the string a had to be put into S in the initial conditions since the recursive rule won't allow us to form that string from λ.

Here is different answer to the same question. It's a little harder to dream up, but the rules are much cleaner. The idea is that if we take two strings with no adjacent a's, we can put them together and be sure to get a new string with no adjacent a's provided we stick either b or c between them. So, we can define the set recursively by (1) $\lambda \in S$ and $a \in S$, and (2) if $x, y \in S$, then $xby, xcy \in S$.

Example 15.6. *Give a recursive definition of the set S of strings over $\Sigma = \{a, b\}$ which contain more a's than b's.*

The idea is that we can build longer strings from smaller ones by (1) sticking two such strings together, or (2) sticking two such strings together along with a b before the first one, between the two strings, or after the last one. That leads to the following recursive definition: (1) $a \in S$ and (2) if $x, y \in S$ then $xy, bxy, xby, xyb \in S$. That looks a little weird since in the recursive rule we added b, but since x and y each have more a's than b's, the two together will have a least two more a's than b's, so it's safe to add b in the recursive rule.

Starting with the initial condition, and then applying the recursive rule repeatedly, we form the following elements of S:

$$a, aa, baa, aba, aab, aaa, baaa, abaa, aaba, baaa, \cdots$$

Example 15.7. *A set, S, of strings over the alphabet $\Sigma = \{a, b\}$ is defined recursively by the rules (1) $a \in S$, and (2) if $x \in S$, then $xbx \in S$. Describe the strings in S.*

Experimenting we find the following elements of S:

$$a, aba, abababa, abababababababa, \cdots$$

It looks like S is the set of strings beginning with a followed by a certain number of ba's. If we look at the number of ba's in each string, we can see a pattern: $0, 1, 3, 7, 15, 31, \cdots$, which we recognize as being the numbers that are one less than the positive integer powers of 2 $(1, 2, 4, 8, 16, 32, \cdots)$. So it appears S is the set of strings which consisting of a followed by $2^n - 1$ pairs ab for some integer $n \geq 0$.

15.3 Exercises

Exercise 15.1. *The set S is described recursively by (1)* $1 \in S$*, and (2) if* $n \in S$*, then* $n+1 \in S$*.*

To what familiar set is S equal?

Exercise 15.2. *Give a recursive definition of the set of positive integers that end with the digits* 17*.*

Exercise 15.3. *Give a recursive definition of the set of positive integers that are not multiples of* 4*.*

Exercise 15.4. *Describe the strings in the set S of strings over the alphabet* $\Sigma = \{a, b, c\}$ *defined recursively by (1)* $\lambda \in S$ *and (2) if* $x \in S$*, then* $axbc \in S$*.*

Exercise 15.5. *Describe the strings in the set S of strings over the alphabet* $\Sigma = \{a, b, c\}$ *defined recursively by (1)* $c \in S$ *and (2) if* $x \in S$ *then* $ax \in S$ *and* $bx \in S$ *and* $xc \in S$*.*

Exercise 15.6. *A* **palindrome** *is a string that reads the same in both directions. For example, aabaa is a palindrome of length five and babccbab is a palindrome of length eight. The empty string is also a palindrome. Give a recursive definition of the set of palindromes over the alphabet*

$\Sigma = \{a, b, c\}$.

A classic palindrome:
A man, a plan, a canal: panama.

15.4 Problems

Problem 15.1. *A set S of integers is defined recursively by the rules:*

(1) $1 \in S$, and (2) If $n \in S$, then $2n + 1 \in S$.

(1) *Is $15 \in S$?*

(2) *Is $65 \in S$?*

Explain your answers.

Problem 15.2. *A set of integers is defined recursively by the rules (1) $0 \in S$, and (2) if $n \in S$, then $2n + 2 \in S$. Give a simple description of the integers in S.*

Problem 15.3. *Give a recursive definition of the set*

$$\{3^n - 3 \mid n \text{ a positive integer}\} = \{0, 6, 24, 78, 240, 726, 2184, \ldots\}.$$

Problem 15.4. *A set, S, of strings over the alphabet $\Sigma = \{a, b, c\}$ is defined recursively by (1) $a \in S$ and (2) if $x \in S$ then $bxc \in S$. List all the strings in S of length seven or less.*

Problem 15.5. *A set, S, of positive integers is defined recursively by the rule:*

(1) $1 \in S$, and (2) If $n \in S$, then $2n - 1 \in S$. List all the elements in the set S.

Problem 15.6. *Give a recursive definition of the set of positive integers that end with the digit 1.*

Problem 15.7. *Give a recursive definition of the set of strings over the alphabet $\Sigma = \{a, b, c\}$ of the form $aaa \cdots abccc \cdots c$. More carefully: zero or more a's followed by a single b followed by the same number of c's as a's.*

Problem 15.8. *Describe the strings in the set S of strings over the alphabet $\Sigma = \{a, b, c\}$ defined recursively by (1) $a \in S$ and (2) if $x \in S$ then $ax \in S$ and $xb \in S$ and $xc \in S$.*

Hint: Your description should be a sentence that provides an easy test to check if a given string is in the set or not. An example of such a description is: S consists of all strings of a's, b's, and c's, with more a's than b's.