Lecture 21. Recursively Defined Structures

Recursive Definition

- Recursion is also useful for many problems that do not involve numbers.
- In general, a recursive definition of a given *object* [1] is made up of two parts:
  - **B.** **Base step** defines the simplest possible object that doesn't depend on anything else, and
  - **R.** **Recursive step** defines more complicated objects recursively that depend on simpler cases.

[1] An object could be a number, a mathematical structure, a function, or almost anything else we want to describe.

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**λ = **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.

  1. Since it is a symbol, e is a palindrome by B2.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.
  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.

  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.
  3. Using R, cec is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.
  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.
  3. Using R, cec is a palindrome.
  4. By B2, a is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.

  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.
  3. Using R, cec is a palindrome.
  4. By B2, a is a palindrome.
  5. By R, aceca is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.

  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')

  B2. Any symbol a is a palindrome.

  R. If x and y are palindromes, then yxy is a palindrome.

- We can build up the palindrome **racecar** from the definition as follows.

  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.
  3. Using R, cec is a palindrome.
  4. By B2, a is a palindrome.
  5. By R, aceca is a palindrome.
  6. By B2, r is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

- **Example 21.1.** A sequence of symbols written together in some order is called a string. λ represents the empty string, and we do not write λ after it has been introduced with another string; for example, **cubs**$\lambda =$ **cubs**. A special kind of string called a palindrome [2] can be defined as follows.
  B1. λ is a palindrome. (Why do we need to define this? Think about forming 'otto')
  B2. Any symbol a is a palindrome.
  R. If x and y are palindromes, then yxy is a palindrome.
- We can build up the palindrome **racecar** from the definition as follows.
  1. Since it is a symbol, e is a palindrome by B2.
  2. Similarly, c is a palindrome.
  3. Using R, cec is a palindrome.
  4. By B2, a is a palindrome.
  5. By R, aceca is a palindrome.
  6. By B2, r is a palindrome.
  7. Using R, racecar is a palindrome.

[2] Any word that is the same forward as backward, such as **racecar** or **HANNAH**

Activity 21.2. Define a set (or collection) X of strings in the symbols 0 and 1 as follows.

B. 0 and 1 are in X.

R1. If x and y are in X, so is $xxyy$.

R2. If x and y are in X, so is xyx.

(a) Explain why the string $01001011 \in X$ using the definition. Build up the string step by step, and justify each step by referring to the appropriate part of the definition.

(b) Use the induction method to prove that, if x is in X, then x has exactly the same number of 0s and 1s.

(c) Find a string in the symbols 0 and 1 that has the same number of 0s and 1s, but is not in X.

Python Code for Palindrome Testing

B1. λ is a palindrome.

B2. Any symbol a is a palindrome.

R. If x and y are palindromes, then yxy is a palindrome.

```python
def isPalindrome(s):
    """Assume s is a string
       Returns True if s is a palindrome; False otherwise.
        Punctuation marks, blanks, and capitalization are
        ignored."""

    def toChars(s):
        s = s.lower()
        letters = ''
        for c in s:
            if c in 'abcdefghijklmnopqrstuvwxyz':
                letters = letters + c
        return letters

    def isPal(s):
        print (' isPal called with', s)
        if len(s) <= 1:
            print (' About to return True from base case')
            return True
        else:
            answer = s[0] == s[-1] and isPal(s[1:-1])
            print (' About to return', answer, 'for', s)
            return answer

    return isPal(toChars(s))
```

```python
def testIsPalindrome():
    print ('Try racecar')
    print (isPalindrome("racecar"))
    print('  ')

    print ('Try AURAK')
    print (isPalindrome('AURAK'))
    print('  ')

    print ("Doc Note: I dissent. A fast never prevents a fatness.I diet on cod.")
    print (isPalindrome("Doc Note: I dissent. A fast never prevents a fatness.I diet on cod."))

testIsPalindrome()
```

Example 21.3. The Koch snowflake fractal. Define a sequence of shapes as follows.

B. $K(1)$ is an equilateral triangle.

R. For $n > 1, K(n)$ is formed by replacing each line segment

of $K(n-1)$ with the shape

Construct $K(1), K(2)$, $K(3)$, etc.

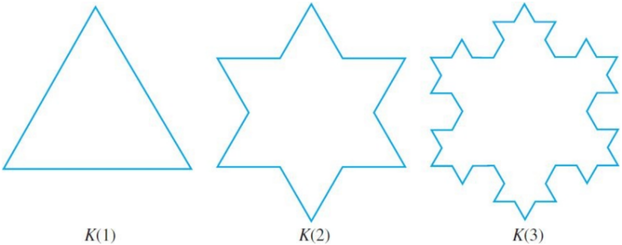

Figure 3.7 The curves $K(1)$, $K(2)$, and $K(3)$.

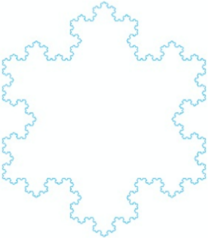

Figure 3.8 The Koch snowflake fractal.