Homework 12

Shyam Sai Bethina

20 October 2021

```java
//1
/**
 * Returns the number of digits of {@code n}.
 *
 * @param n
 *            {@code NaturalNumber} whose digits to count
 * @return the number of digits of {@code n}
 * @ensures numberOfDigits = [number of digits of n]
 */
private static int numberOfDigits(NaturalNumber n) {
    int count = 0;
    NaturalNumber temp = new NaturalNumber2(n);
    temp.divideBy10();
    if (!temp.isZero()) {
        count++;
        numberOfDigits(temp);
    }
    return count;
}

//2
/**
 * Returns the sum of the digits of {@code n}.
 *
 * @param n
 *            {@code NaturalNumber} whose digits to add
 * @return the sum of the digits of {@code n}
 * @ensures sumOfDigits = [sum of the digits of n]
 */
private static int sumOfDigits(NaturalNumber n) {
    int sum = 0;
    NaturalNumber temp = new NaturalNumber2(n);
    int lastDigit = temp.divideBy10();
    if (!temp.isZero()) {
        sum += lastDigit;
        sumOfDigits(temp);
    }
    return sum;
}
```

```java
//3
/**
 * Returns the sum of the digits of {@code n}.
 *
 * @param n
 *            {@code NaturalNumber} whose digits to add
 * @return the sum of the digits of {@code n}
 * @ensures sumOfDigits = [sum of the digits of n]
 */
private static NaturalNumber sumOfDigits(NaturalNumber n) {
    NaturalNumber sum = new NaturalNumber2();
    NaturalNumber temp = new NaturalNumber2(n);
    int lastDigit = temp.divideBy10();
    if (!temp.isZero()) {
        NaturalNumber tempLastDigit = new NaturalNumber2(lastDigit);
        sum.add(tempLastDigit);
        sumOfDigits(temp);
    }

    return sum;
}

//4
/**
 * Divides {@code n} by 2.
 *
 * @param n
 *            {@code NaturalNumber} to be divided
 * @updates n
 * @ensures 2 * n <= #n < 2 * (n + 1)
 */
private static void divideBy2(NaturalNumber n) {
    int lastDigit = n.divideBy10();
    if(!n.isZero()) {
        if(lastDigit % 2 != 0) {
            lastDigit = lastDigit / 2 + RADIX/2;
        } else {
            lastDigit /= 2;
        }

        divideBy2(n);
        n.multiplyBy10(lastDigit);
    }


}
```

```java
//5
/**
 * Checks whether a {@code String} is a palindrome.
 *
 * @param s
 *            {@code String} to be checked
 * @return true if {@code s} is a palindrome, false otherwise
 * @ensures isPalindrome = (s = rev(s))
 */
private static boolean isPalindrome(String s) {
    Boolean isPalindrome = false;
    int count = 0;
    String firstChar = s.substring(count, count + 1);
    String lastChar = s.substring(s.length() - 2 - count,
            s.length() - 1 - count);

    if (firstChar.equals(lastChar) && count < s.length() / 2) {
        count++;
        isPalindrome(s);
        isPalindrome = true;
    }

    return isPalindrome;
}
```