CSE 20 Homework 6
Due: Saturday November 18, 2017 at 11pm

TOPICS Sets and Induction

READING Rosen Sections 5.1, 5.2, 5.3

KEY CONCEPTS Mathematical induction, recursive definitions, strong induction, structural induction, basis step, induction hypothesis, induction step, strings.

---

1. (**6 points**) Give recursive definitions of each of the following sets. Remember to include a basis step and a recursive step. Give a brief justification of why your definition works.

(a) The set of odd integers (both positive and negative).

(b) The set of coordinates on the integer grid $\mathbb{Z} \times \mathbb{Z}$ where the sum of the coordinates is even.

This example might help you get started: Let's give a recursive definition of the set of positive multiples of 3. *We'll call the set* S.

Basis step*:* $3 \in S$

Recursive step*: If* x, y *are both in* S *then* $x + y \in S$.

The positive multiples of 3 *are* $3, 6, 9, 12, \ldots$. *We can build each one from earlier ones by adding* 3 *over and over again.*

2. (**12 points**) The **Kleene star** is an operation which, on input a set of strings A, outputs a new set of strings (that we call A^*) which is defined recursively by

Basis step: $\lambda \in A^*$

Recursive step: If u is in $A^*$and v is in A, then $uv \in A^*$

Note: when we write λ *we mean the empty string (see definition on page 349) and when we write* uv *we mean the* **string concatenation** *of* u *and* v. *The formal definition is Definition 2 on page 350.*

For each of the following sets, give an example of a nonempty bit string that **is** an element of the set, and give an example of a nonempty bit string that **is not** an element of the set (make sure to label which is which). If it is **not possible** to give these two examples, say so and explain why.

(a) $\{01\}^*$

(b) $\{0, 11, 000\}^*$

(c) $\{0\}^* \cup \{1\}^*$

(d) $\{001\}^* \cap \{001001\}^*$

(e) $\emptyset^*$

This example might help you get started: $\{1\}^*$. *This is the set defined recursively by* $\lambda \in \{1\}^*$, *and if* $u \in \{1\}^*$ *and* $v \in \{1\}$ *then* $uv \in \{1\}^*$. *Since there's only one element in* $\{1\}$, *the only possible value for* v *is* 1. *In other words, as we recursively build up this set, we start (at the basis step) with just* $\{\lambda\}$. *At the first application of the recursive step, the only choice for* u *is the one element that's been added to the set so far:*

$u = \lambda$. Therefore, $uv = \lambda 1 = 1$. So, at the end of this step, the set currently looks like $\{\lambda, 1\}$. At the next application of the recursive step, $u \in \{1\}^*$ means $u = \lambda$ or $u = 1$; still, $v = 1$. Thus, the possible values of uv are $\lambda 1 = 1$ and 11. After adding these to the set being built, the set at this stage is $\{\lambda, 1, 11\}$. Continuing on, we see that $\{1\}^* = \{\lambda, 1, 11, 111, \ldots\} = \{1^i \mid i \in \mathbb{N}\}$ (where we use the convention that, for any string x, $x^0 = \lambda$). Therefore, an example of a nonempty bit string that **is** *an element of this set is* 1111. *An example of a nonempty bit string that* **is not** *an element of this set is* 0.

3. (**8 points**)

(a) Give a recursive definition of the function $zeros(s)$, which counts the number of zeros in a bit string $s \in \{0, 1\}^*$.

Hint: the domain of this function is $\{0, 1\}^*$ and its codomain is the set of nonnegative integers $\mathbb{N}$. For example, $zeros(1101) = 1$ and $zeros(0010) = 3$.

(b) Use structural induction to prove that $zeros(st) = zeros(s) + zeros(t)$.

Your proof will need to use the definition you came up with in part (a).

Note that st refers to the concatenation of the strings s and t; on the RHS the $+$ is addition of integers.

Hint: you may find the recursive definition of **string concatenation**, *Definition 2 on page 350, useful.*

4. (**12 points**) The set of **full binary trees** is defined recursively in Definition 5 on page 353:

Basis step: There is a full binary consisting only of a single vertex r.

Recursive step: If T_1 and T_2 are disjoint full binary trees, there is a full binary tree, denoted by $T_1 \cdot T_2$, consisting of a (new) root r together with edges connecting this root to the roots of the left subtree T_1 and the right subtree T_2.

The book includes these pictures of the full binary trees that can be built up by applying the recursive step one or two times.

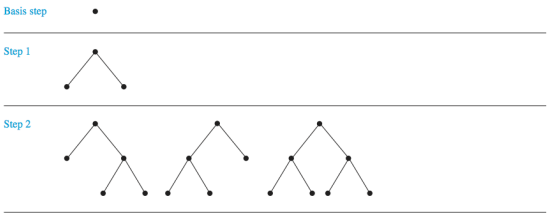

FIGURE 4 **Building Up Full Binary Trees.**

(a) Draw all the binary trees that could be built up by applying the recursive step **three** times.

(b) Come up with a formula for the **maximum** number of nodes (vertices) in a full binary tree that is built up by applying the recursive step n times. Prove your formula using mathematical induction.