CSE 20 Homework 6
Due: Sunday February 26, 2017 at noon

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. (**12 points**) Suppose that $P(n)$ is a predicate (also known as a propositional function). Determine for which nonnegative integers n the statement $P(n)$ must be true in each of the following cases.

(a) $P(2)$ is true; for all nonnegative integers n, if $P(n)$ is true, then $P(n+2)$ is true.

(b) $P(0)$ is true; for all nonnegative integers n, if $P(n)$ is true, then $P(n+3)$ is true.

(c) $P(0)$ and $P(1)$ are true; for all nonnegative integers n, if $P(n)$ and $P(n+1)$ are true, then $P(n+2)$ is true.

(d) $P(0)$ is true; for all nonnegative integers n, if $P(n)$ is true, then $P(n+2)$ and $P(n+3)$ are true.

As an example to get started, let's look at the case where $P(0)$ is true; for all nonnegative integers n, if $P(n)$ is true, then $P(n+1)$ is true. This is the setup for mathematical induction so we can conclude that for **all** *nonnegative integers n, $P(n)$ is true.*

For full credit, describe the set of values of n for which $P(n)$ is guaranteed to be true and **informally** *justify your answer. You do not need a precise proof for this question.*

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

$$\lambda \in A^*$$
$$\text{If } u \text{ is in } A^* \text{ and } v \text{ is in } A, \text{ then } uv \in A^*$$

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. If it is **not possible** to give these two examples, say so and explain why.

(a) $\{00\}^*$

(b) $\{1, 11, 111\}^*$

(c) $\{00, 111\}^*$

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

(e) $\{101\}^* \cap \{10101\}^*$

(f) $\emptyset^*$

An example to 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.

And another example: let's consider $\{0, 1\}^*$. *This is the set of all possible bit strings. An example of a nonempty bit string that* **is** *an element of this set is* 101. *It is* **not possible** *to give an example of a nonempty bit string that is not an element of this set because all bit strings are in this set.*

3. (**8 points**) A **palindrome** is a string that reads the same forward and backward. For example, 11011 and 1 are both palindromes; 10 is not a palindrome. Give a recursive definition of the set of bit strings that are palindromes.

Hint: you can use the recursive definition of the set of all strings in Definition 1 on page 349 or the recursive definition of the Kleene star in question 2 as a guide to how to write out your recursive definition.

Another hint: how many elements do you need to consider in the basis step?

4. (**18 points**)

(a) Give a recursive definition of the function $ones(s)$, which counts the number of ones 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* $\mathbb{N}$.

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

Bonus Question - not for credit Prove (by structural induction on strings) that in a bit string, the string 01 occurs at most one more time than the string 10.

For example, if we consider the string 001010011, *then the string* 01 *occurs three times (starting at position 2, 4, 7 if we index positions starting at index 1) and the string* 10 *occurs twice (starting at position 3 and position 5). Since three is at most one more than two, the statement is confirmed for this example.*

As another example, let's consider the string 0010. *In this string,* 01 *and* 10 *each occur once. Since they occur the same number of times, the statement is confirmed for this example too.*