

198:205
Discrete Structures
Professor McCarty

Homework 10

This single problem encompasses all the main ideas in Sections 4.3, 4.4 and 4.5 of the Rosen text. Note, however, that the recursive clause in the definition of a string is slightly different. In this problem, a new symbol is added to a string by a prefix operation; whereas, in the text, a new symbol is added by a postfix operation.

1. Recall the definition of Σ^* from your class notes. If Σ is any finite set of symbols and λ is the empty string, then the set Σ^* is defined as follows:
 - $\lambda \in \Sigma^*$
 - If $\sigma \in \Sigma$ and $x \in \Sigma^*$ then $\sigma x \in \Sigma^*$
 - Σ^* is the intersection of all sets (i.e., the smallest set) satisfying these properties.

Suppose we want to search an arbitrary string in Σ^* to locate an occurrence (if there is one) of an arbitrary symbol in Σ . Define the function $search : \Sigma \times \Sigma^* \mapsto \Sigma^*$ as follows:

- For every $\sigma \in \Sigma$, $search(\sigma, \lambda) = \lambda$,
- For every $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$, and for every $x \in \Sigma^*$,

$$\begin{aligned} search(\sigma_1, \sigma_2 x) &= \sigma_2 x, \text{ if } \sigma_1 = \sigma_2 \\ &= search(\sigma_1, x), \text{ if } \sigma_1 \neq \sigma_2 \end{aligned}$$

Intuitively, $search(\sigma, x)$ returns the empty string if σ does not occur in x ; otherwise, if σ does occur in x , then $search(\sigma, x)$ returns a *suffix* of x that begins with σ .

Let $\Sigma = \{a, b, c\}$, and answer the following questions:

- (a) (5 points) Write out the computation of $search(a, cbabab)$.
- (b) (5 points) Write out the computation of $search(a, cbc b)$.

- (c) (10 points) Prove that *search* is a well-defined function everywhere on its domain.
2. Let Σ and Σ^* be the same as in the previous question. Define the *suffix* relation $\mathbf{S} \subseteq \Sigma^* \times \Sigma^*$ as follows:
- For every $x \in \Sigma^*$, $(x, x) \in \mathbf{S}$.
 - If $(x, y) \in \mathbf{S}$ and $\sigma \in \Sigma$, then $(x, \sigma y) \in \mathbf{S}$.
 - \mathbf{S} is the intersection of all sets (i.e., the smallest set) satisfying these properties.

Intuitively, $(x, y) \in \mathbf{S}$ if and only if the string x is a suffix of the string y . Answer the following questions:

- (a) (5 points) Show that $(ab, cbabab) \in \mathbf{S}$.
- (b) (5 points) Show that $(\lambda, ab) \in \mathbf{S}$.
- (c) (5 points) Show that $(ab, \lambda) \notin \mathbf{S}$.
- (d) (5 points) Prove, for every $x \in \Sigma^*$, that if $x \neq \lambda$ then $(x, \lambda) \notin \mathbf{S}$.
3. Let Σ , Σ^* , *search* and \mathbf{S} be the same as in the previous two questions. We will now establish a connection between the inductive definition of *search* and the inductive definition of \mathbf{S} .

- (a) (15 points) Prove the following by structural induction (on what?):

Proposition. For every $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$, and for every $x \in \Sigma^*$ and $z \in \Sigma^*$,

if $search(\sigma_1, x) = \sigma_2 z$ **then** $(\sigma_2 z, x) \in \mathbf{S}$ and $\sigma_1 = \sigma_2$.

- (b) (15 points) Prove the following by structural induction (on what?):

Proposition. For every $\sigma \in \Sigma$ and for every $x \in \Sigma^*$,

if $search(\sigma, x) = \lambda$ **then** $\neg \exists z [z \in \Sigma^* \text{ and } (\sigma z, x) \in \mathbf{S}]$.

Intuitively, these propositions tell us something about the correctness of our search procedure, as you will see in the following question.

4. Suppose we convert our recursive definition of *search* into a recursive procedure. (As a notational convention, if a string is written "abc", then the empty string would be written ""). The pseudocode for $search(c, s)$, where c is a character and s is a string, might be written as follows:

```

procedure search(c,s)
  if s = "" then return ""
  else if head(s)=c then return s
  else return search(c,tail(s))

```

The procedures `head(s)` and `tail(s)` correspond to the functions *head* and *tail* as defined in class.

- (a) (10 points) Use the two propositions in the previous question to argue that this procedure is correct. (Hint: You should use the concept of a *suffix* to specify the expected result of the search, and assume that this concept is correctly formalized by the relation **S**. Then show that the procedure and the specification coincide.)
5. Suppose, instead, that we convert our recursive definition of *search* into an iterative procedure. The pseudocode for `search(c,s)` might then be written as follows:

```

procedure search(c,s)
  t:=s
  while t != ""
    begin
      if head(t)=c then return t
      else t:=tail(t)
    end
  return ""

```

We would like to show that this version of the procedure is also correct.

- (a) (10 points) Consider the following proposition:

$$\exists y[(y, s) \in \mathbf{S} \text{ and } \text{head}(y)=c] \text{ iff}$$

$$\exists y[(y, t) \in \mathbf{S} \text{ and } \text{head}(y)=c]$$

Show that this proposition is an invariant for the `while` loop in `search(c,s)`.

- (b) (10 points) Use this loop invariant to show that `search(c,s)` is correct, i.e., that it coincides with the specification that you formalized above using the relation **S**.