CSCI 432 Handout 07: Decrementing Functions

Name: ________________________________

Collaborators: ________________________________

5 February 2026

Definitions

Definition 1 (State Space, $\mathcal{S}$)**.** *A* state *is a snapshot in the execution of an algorithm. It includes all states/values of all variables, stack traces, the number of times a loop has executed, etc. Think of this as everything that you can see uising a breakpoint when running your code through a debugger. Note that some variables might be implicit (e.g., number of times a while loop has executed).*

The state space *of a program is the set of all states that are realized during the execution of the program. We call this a* space *and not a* set *as we can add edges between "adjacent" states to give the set more structure (namely, that of a digraph). Here, we just care about the set of all states though.*

Definition 2 (Well-Ordered Set)**.** *A* well-ordered set *is a partially ordered set (poset) such that every subset has a minimum or least element. Mathematically, $(S, \leq)$ is a well-ordered set iff for all $S' \subset S$, $\exists! e \in S'$ such that for all $s \in S'$, $e \leq s'$.*

Our favorite well-ordered sets are $\mathbb{N} := \{0, 1, 2, \ldots\}$ and $\mathbb{Z}_+ := \{1, 2, 3, \ldots\}$.

Decrementing Functions

A decrementing function is used to prove that an algorithm (specifically, a loop or a recurrence) terminates. We use them in two ways: (1) when doing back-of-hand calculations to ensure that our algorithm will terminate before doing a proper runtime analysis; (2) when it is difficult to prove exact runtime.

Definition 3 (Decrementing Function)**.** *A decrementing function is a function $D \colon \mathcal{S} \to \mathbb{N}$ that satisfies the following properties:*

(1) Each time a recursive call is made, the function strictly decreases between the top of the parent call to the top of the child call.

(2) Each time a loop is re-entered, the function is strictly less than the last time it entered that loop.

If such a function exists, then our algorithm terminates! Can you explain why this is true?

Practice

Provide decrementing functions for the following algorithms:

1. Suppose the algorithm has a single for loop, which appears as follows: `for i=10; i >0; i--`. What is the decrementing function?
   Try it!

2. Write a for loop that iterates through an array A of length n starting at $A[1]$ and checking each index in order until $A[n]$, in order to find the maximum value in the array. Then, provide the decrementing function for that loop.
   Try it!

3. Suppose there is a perfect infinite neighborhood, where the post office is at $(0, 0)$ and for every $x, y \in \mathbb{N}$, there is a house at (x, y). If you a person in this neighborhood can follow simple instructions (e.g., walk one block north/south/east/west), write a while loop that will give directions from a given house to the post office. Then, prove that your algorithm terminates.
   Try it!

4. Bubble sort. (Note: there are two loops here! On first attempt, just worry about the loops independently and come up with two functions.)