**CS 4820, Spring 2019** **Homework 9, Problem 2**
Name: Kevin Klaben NetID: kek228

**(2)** *(10 points)* As you probably know, buffer overflows are a common security flaw in computer programs that can be exploited by attackers to overwrite areas of memory that should have been protected from access. This exercise explores why it's so difficult to perform static analysis of programs to discover potential buffer overflows. In each of the sub-questions below, we use the term *buffer program* to refer to an SJava program whose base function contains a variable named "buffer" of string type. The *length of the buffer* refers to the number of characters in this string.

For each of the following decision problems, determine whether the problem is decidable, recognizable but not decidable, or not recognizable. Substantiate your answer with a proof. If one part of your solution uses material from an earlier part, it is fine to refer back to that earlier part of your solution rather than repeating the material.

**(2a)** *(5 points)* $\mathcal{L}_{\text{A}}$ is the set of all ordered triples $\langle M, x, k\rangle$ such that $M$ is a buffer program, and when $M$ runs on input $x$ the length of the buffer exceeds $k$ at least once during its execution.

**(2b)** *(5 points)* $\mathcal{L}_{\text{B}}$ is the set of all ordered pairs $\langle M, x\rangle$ such that $M$ is a buffer program, and when $M$ runs on input $x$ the length of the buffer grows unboundedly large. (Meaning: $M$ runs forever on input $x$, and for all $k$ there is a time during its execution when the length of the buffer exceeds $k$.)

(2a) Claim: $\mathcal{L}_{\text{A}}$ is recognizable.

Proof: by construction of a program which recognizes inputs. A program which checks the buffer after each step for length of the buffer. If the length of the buffer is less then k then the program continues executing, while if it is greater than k, it accepts and terminate the program. Thus, as the program will terminate on all programs if the buffer becomes longer than k, this language is recognizable.

Claim: $\mathcal{L}_{\text{A}}$ is not decidable

Proof: As was seen in class, we can apply Rice's Theorem to this language. Rice's theorem states that all languages which are non-trivial are unrecognizable. The language defined by Claim: $\mathcal{L}_{\text{A}}$ is not decidable by Rice's theorem as $\mathcal{L}_{\text{A}}$ is defined based on a nontrivial property for which the program M must meet. As $\mathcal{L}_{\text{A}}$ is defined non-trivially, it is undecidable.

(2b) Claim: $\mathcal{L}_{\text{B}}$ is unrecognizable.

Proof: As we know the language of all programs which halt is recognizable and its complement is unrecognizable. We can reduce from the complement of the halting problem to this problem in which we are in search of programs which have no bound to their buffers growth. These programs cannot halt as thie would cause the buffers to cease to grow thus, this language like that of the complement of the halting problem is unrecognizable.

Claim: $\mathcal{L}_{\text{B}}$ is not decidable.

Proof: As was seen in class, we can apply Rice's Theorem to this language. Rice's theorem states that all languages which are non-trivial are unrecognizable. The language defined by Claim: $\mathcal{L}_{\mathrm{B}}$ is not decidable by Rice's theorem as $\mathcal{L}_{\mathrm{B}}$ is defined based on a nontrivial property for which the program M must meet. As $\mathcal{L}_{\mathrm{B}}$ is defined non-trivially, it is undecidable.