# Methods necessary for string classes in object-oriented programming languages

## 1 INTRODUCTION

In programming, a string is a primitive type designed to provide an interface for an ordered set of characters. However, standard library implementations of this interface vary in different languages. Thus, it is clear that the programming community has not decided on the optimal implementation of a string class. Therefore, this paper aims to determine a minimalistic set of string functions sufficient for a string class.

## 2 SET BUILDING

### 2.1 Necessary and sufficient functions

The memory tape of the Turing machine can be represented as three strings: parts of the memory tape right and left from the pointer and the memory cell with the pointer. Three operations are necessary for a Turing machine: move the head to the left, move the head to the right, and write a symbol to the cell. Therefore, the following methods are necessary:

(1) concatenate - concatenate two strings
(2) char-at - returns the character at a given position as a string

It is easy to see that these methods are sufficient to implement Turing machine commands. Therefore, all computable algorithms, including string-related algorithms, can be described through strings.

### 2.2 Extending the set

However, such an approach is not practical, and other methods need to be included. For this purpose, the following algorithm for set construction is proposed:

(1) Functions char-at and concatenate are added to the set
(2) The function that is
    (a) Can not be implemented in two or fewer instructions using functions already in set
    (b) Can be implemented in the least amount of instructions among all remaining functions
    is added to the set
(3) The second step is repeated until no fitting functions remain

Author's address: