

# Implementation and Evaluation of the FunArray Abstract Domain in Java

## Bachelor's Thesis

Maximilian Hofstetter

Software and Computational Systems Lab  
Ludwig-Maximilians-Universität München

### 1 Introduction

### 2 Scope of Application

Lorem ipsum dolor sit amet

### 3 Limitations

#### 3.1 Adjacent potentially empty segments

When expanding a segment by inserting a value adjacent to it, formerly adjacent and potentially empty segments will be joined with their nearest neighbour until a non-empty segment is reached. Let us consider the FunArray  $A : \{a\} x \{b\} y \{c\}? z \{d\}$  for example. It is unknown whether its second segment containing the value  $y$  is empty or not and thus whether  $b = c$  or  $b < c$  is true.

When trying to expand the first segment by inserting its value  $x$  at index  $b$ , the resulting FunArray could either be  $\{a\} x \{b\} c x \{b+1\}? z \{d\}$ , if the  $y$ -segment was empty, or alternatively  $\{a\} x \{b\} x \{b+1\} y \{c\}? z \{d\}$ , if it was non-empty. Since the position of the  $\{c\}$ -bound cannot be decided, it must be discarded altogether. After the insertion,  $A$  becomes  $A' : \{a\} x \{b\} x \{b+1\}? y \sqcup z \{d\}$  and all information concerning the variable  $c$  is lost.

Analyses of programs utilising segments in their arrays with adjacent and potentially empty segments have to discard non-trivial information. Thus the FunArray's approach for handling potentially empty adjacent segments limits the number of programs for which an analysis can achieve a non-trivial result.

Consider Algorithm 1, Dijkstra's Dutch national flag algorithm, for example. Even though the analysis correctly determines the Array to be described as  $\{0\} [-\infty, -1] \{r\}? \top \{w\}? [0, 0] \{b\}? [1, \infty] \{|A|\}? after a single loop pass, it loses all its information in the second pass, because the determined segments are all potentially empty and extending their neighbours discards them altogether. After the second loop pass the analysis yields  $\{0\} \top \{|A|\}$ . Basically no information can be gained$

```

 $r \leftarrow 0$ 
 $w, b \leftarrow \text{length of } A - 1$ 
while  $r \leq w$  do
    if  $A[w] \leq -1$  then
         $temp \leftarrow A[r]$ 
         $A[r] \leftarrow A[w]$ 
         $A[w] \leftarrow temp$ 
         $r \leftarrow r + 1$ 
    if  $A[w] = 0$  then
         $w \leftarrow w - 1$ 
    if  $A[w] \geq 1$  then
         $temp \leftarrow A[w]$ 
         $A[w] \leftarrow A[b]$ 
         $A[b] \leftarrow temp$ 
         $w \leftarrow w - 1$ 
         $b \leftarrow b - 1$ 

```

**Algorithm 1:** Dijkstra’s proposition for an algorithm for solving the Dutch national flag problem [1].

## References

1. Edsger Dijkstra. *A Discipline of Programming*, chapter 14, pages 111–116. Prentice-Hall, 1976.