**Comp301 – Project 3 Report**

1. Workload:

Hasan Can Aslan: Array - Report

Necla Mutlu: Queue.

Fırat Tamur: Stack

1. All parts are working with given test cases.

1. Implementation:

Array: In the array, arr-val has only one field called “arr”, which is a scheme list of ref-val’s.

new-array: Using the helper-new-array function, newarray-exp creates a scheme list of ref-val’s, then it makes an arr-val from this scheme list.

update-array: update-array-exp calls helper-update-array function with the underlying scheme list of the given arr-val expression. helper-update-array traverses this scheme list until the index that needs to be changed. When it finds the index, it takes the ref-val at that index and change the value stored in this reference.

read-array: read-array-exp calls helper-read-array function with the underlying scheme list of the given arr-val expression. helper-read-array traverses this scheme list until the target index. When it finds the ref-val at that index, it derefs and returns it.

Stack: In the stack, we have an underlying array with 1001 elements.

newstack: newstack-exp creates and evaluates a newarray-exp, which will create an array with 1001 elements. Initially, -1 was stored in each cell of the array.

push: stack-push-exp finds the first index that stores -1 in the given stack expression (indeed an array) using first-empty-index-in-array function. Then provides this index to helper-update-array function with the underlying array and new value of the cell. It changes the value of the cell.

pop: stack-pop-exp finds the size of the stack. The last element is stored in the index (size -1). After getting the ref-val at that position in the underlying array and dereferencing it, stack-pop-exp uses helper-update-array function to update the element at position (size – 1) with -1. So that it makes the cell an empty cell.

top: stack-top-exp finds the size of the stack. The last element is stored in the index (size -1). It gets the ref-val at that position in the underlying array, dereferences it, and returns.

empty?-stack: It uses array-size function to find the size of the underlying array. array-size function uses helper-empty-array to check whether the array is empty. helper-empty-array function checks whether there is any element whose value is different than -1. If there is, it returns false, otherwise it returns true. array-size function recursively finds the size of the array if helper-empty-array returns false.

size: stack-size-exp uses array-size function to find the number of elements whose value is not -1.

print: print-stack-exp uses helper-print-array to print the elements in the stack. helper-print-array has its own helper function in it called helper. What this helper does is that it accumulates all elements of stack that are different than -1 in a list. When there is no other element, it joins the element of this list by spaces and returns the resulting string. helper-print-array displays this string.

Queue:

newqueue: newqueue-exp creates and evaluates a newarray-exp, which will create an array with 1001 elements. Initially, -1 was stored in each cell of the array.

push: queue-push-exp finds the first index that stores -1 in the given queue expression (indeed an array) using first-empty-index-in-array function. Then provides this index to helper-update-array function with the underlying array and new value of the cell. It changes the value of the cell.

pop: queue-pop-exp, first checks whether the given queue is empty by using helper-empty-array function for the underlying array of the queue. If it is empty it returns (num-val -1). Otherwise, it takes the first reference in the underlying array and dereferences it. Then it stores it as the value to pop. After that, it calls helper-pop-queue function with the underlying array. helper-pop-queue function moves every element to left by one using helper-update-array function and updates the value of the last non-empty cell with -1. So that in the next pop/top operations everything works properly without keeping a pointer to the first/last element of the queue. The pop value is returned afterwards.

top: first checks whether the given queue is empty by using helper-empty-array function for the underlying array of the queue. If it is empty it returns (num-val -1). Otherwise, it takes the first reference in the underlying array, dereferences it, and returns it.

empty?-queue: It uses array-size function to find the size of the underlying array. array-size function uses helper-empty-array to check whether the array is empty. helper-empty-array function checks whether there is any element whose value is different than -1. If there is, it returns false, otherwise it returns true. array-size function recursively finds the size of the array if helper-empty-array returns false.

size: queue-size-exp uses array-size function to find the number of elements whose value is not -1.

print: print-queue-exp uses helper-print-array to print the elements in the stack. helper-print-array has its own helper function in it called helper. What this helper does is that it accumulates all elements of queue that are different than -1 in a list. When there is no other element, it joins the element of this list by spaces and returns the resulting string. helper-print-array displays this string.