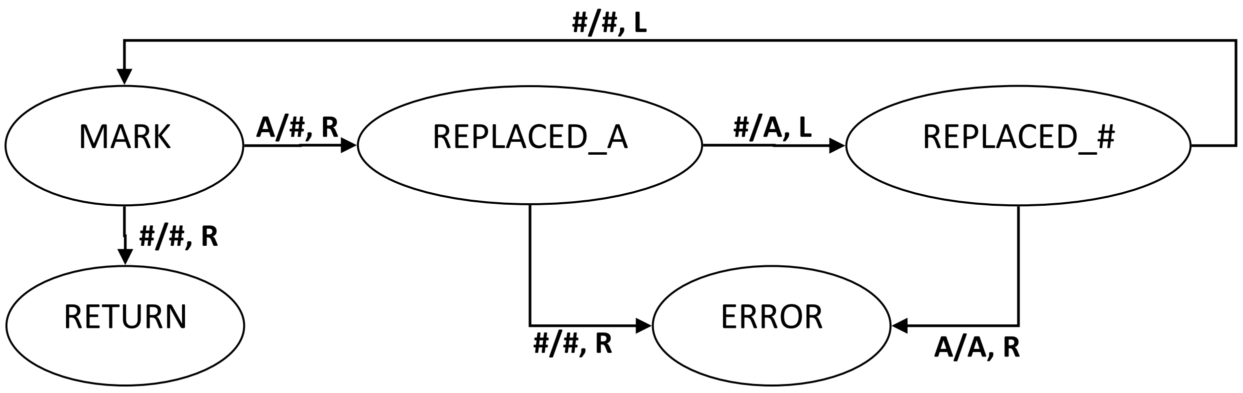
CPS2001 – Assignment 1

Task 1

The Turing Machine

The below Turing Machine is programmed to perform the addition of two integers. The starting state is labelled as **MARK**, while the two halting states are labelled as **RETURN** (for when addition terminates successfully) and **ERROR** (for when the representation was invalid).



The Representation Used

The representation chosen uses an alphabet made up of the symbols **A** and **#**, where **A** indicates a value of **1**, while **#** is used to group sequences of **A**s. Consider the following general addition:

$$\boldsymbol{N+M}$$

Such a general addition will be represented by two sequences of **A**s separated by a **#**. The first sequence represents the first (left) operand, while the second sequence represents the second (right) operand. The amount of **A**s in a particular sequence is equal to the respective operand's value. Hence, the sequences will contain an 'N' and an 'M' amount of **A**s, respectively.

The ends of the representation are indicated by a **#** such that the leftmost and rightmost symbols are **#**s. For a general addition, the starting representation is of the following form:

$$\#\underbrace{\boldsymbol{AAA}\ldots}_{N\ times}\#\underbrace{\boldsymbol{AAA}\ldots}_{M\ times}\#$$

Similarly, the final representation is of the following form:

$$\#\#\underbrace{\boldsymbol{AAA}\ldots}_{(N+M)\ times}\#$$

The machine's starting position will be the first symbol on the left of the middle **#**. In the case where $N = 0$, this will point to the left **#**. Otherwise, the starting position will point to an **A**.

Dry Runs

In the following dry-runs, the representation described previously will be used. Additionally, an underline will indicate the current position of the machine on the tape.

i) 1+5

State	Current Representation	Description of Next Transition
MARK	#A#AAAAA#	Found **A**; Replace with **#** and move to the right (A/#,R)
REPLACED_A	###AAAAA#	Found **#**; Replace with **A** and move to the left (#/A,L)
REPLACED_#	##AAAAAA#	Found **#**; Replace with **#** and move to the left (#/#,L)
MARK	##AAAAAA#	Found **#**; Replace with **#** and move to the right (#/#,R)
RETURN	##AAAAAA#	Finished (No further transitions)

ii) 4+2

State	Current Representation	Description of Next Transition
MARK	#AAAA#AA#	Found **A**; Replace with **#** and move to the right (A/#,R)
REPLACED_A	#AAA##AA#	Found **#**; Replace with **A** and move to the left (#/A,L)
REPLACED_#	#AAA#AAA#	Found **#**; Replace with **#** and move to the left (#/#,L)
MARK	#AAA#AAA#	Found **A**; Replace with **#** and move to the right (A/#,R)
REPLACED_A	#AA##AAA#	Found **#**; Replace with **A** and move to the left (#/A,L)
REPLACED_#	#AA#AAAA#	Found **#**; Replace with **#** and move to the left (#/#,L)
MARK	#AA#AAAA#	Found **A**; Replace with **#** and move to the right (A/#,R)
REPLACED_A	#A##AAAA#	Found **#**; Replace with **A** and move to the left (#/A,L)
REPLACED_#	#A#AAAAA#	Found **#**; Replace with **#** and move to the left (#/#,L)
MARK	#A#AAAAA#	Found **A**; Replace with **#** and move to the right (A/#,R)
REPLACED_A	###AAAAA#	Found **#**; Replace with **A** and move to the left (#/A,L)
REPLACED_#	##AAAAAA#	Found **#**; Replace with **#** and move to the left (#/#,L)
MARK	##AAAAAA#	Found **#**; Replace with **#** and move to the right (#/#,R)
RETURN	##AAAAAA#	Finished (No further transitions)

Task 2

For this task, the set of functions providing various set operations were implemented in a **sets.c** file included in the digital submission of the assignment along with the files **main.c** and **sets.h**, all in a folder named **src**. The following are two assumptions that were taken for this task:

Assumption 1.	No set_element will be used after it is destroyed.
Assumption 2.	No NULL pointers will be passed as a parameter to any function.

Note, however, that some functions are still able to process NULL pointers without a problem.