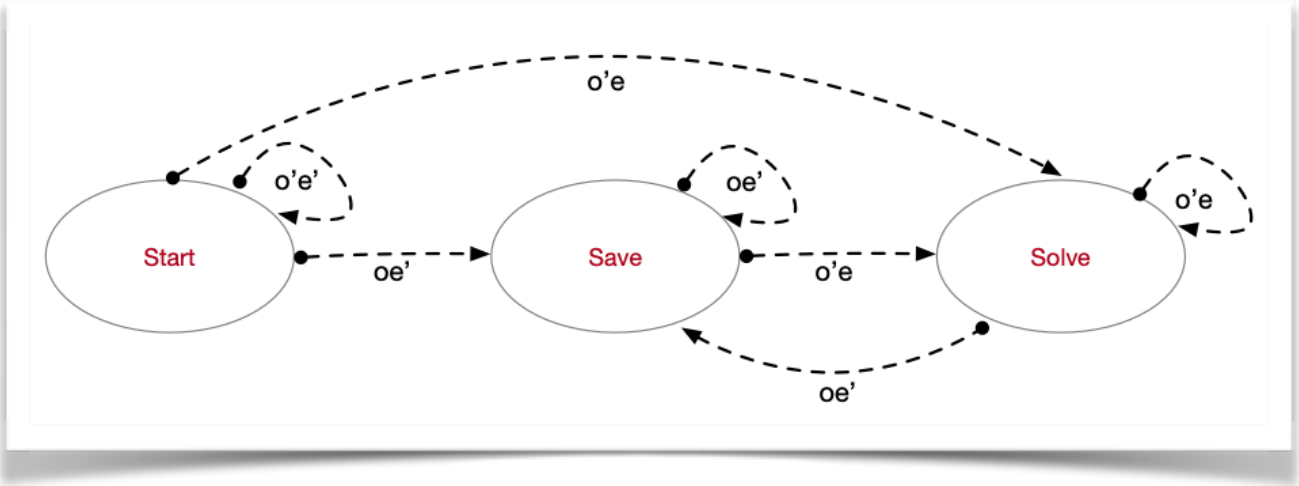
James J. Kim
SER232 - Summer Term B

Final Project - Part 1 Design Document



1. **Start State:** Utilizes user's initial value, we're now waiting for user to choose their operator (+,-,*,/) to transition to *Input* state or *Solve state.*
2. **Save State**: Saves the first user input value and waiting for the second value. Regardless if new value is selected, clicking on "Equal" button will transition to the *Solve* state. If the user decides to click the operator again, it will consume the number waiting at the Start state and use that value as the first number.
3. **Solve State**: Calculates the user input values and stores the value. It will transition back to *Save* state when an Operator is pressed.

** I thought about adding the secondary wait state after save but I felt that when you transition from Save state to Solve state, FSM would've already saved the first number and the second number that's waiting.

Key
ø = Operator (+, -, *, /)
e = Equal
y = Output 0
x = Output 1

Truth Table

S_1	S_0	ø	e	x	y	nextStep
0	0	0	0	0	0	Start (00)
0	0	0	1	0	1	Solve (10)
0	0	1	0	1	0	Save (01)
0	0	1	1	0	0	Start (00)
0	1	0	0	0	0	Save (01)
0	1	0	1	0	0	Solve (10)
0	1	1	0	0	0	Save (01)
0	1	1	1	0	0	Save (01)
1	0	0	0	0	0	Solve (10)
1	0	0	1	0	0	Solve (10)
1	0	1	0	0	0	Save (01)
1	0	1	1	0	0	...
1	1	0	0	0	0	...
1	1	0	1	0	0	...
1	1	1	1	0	0	...

All the transitions that was listed in the FSM. Used boolean algebra to limit to canonical terms.
nextStep = ø'e + ø e' + ø'e + ø e' + ø'e' + ø e' + ø'e
nextStep = ø'e + ø e'+ ø e' + ø'e' + ø e'
nextStep = ø e + ø e' + ø'e' + ø'e