I was able to get the universal turing machine to work on an a star TM and the triple TM. I tested them with the basic inputs, i.e. “a”, “aa”, “”, “b”, “ab” for a star and “abc”, “ac”, “ab”, “bc”, “aabbcc” for triple TM, and got the expected/correct outputs. I didn’t make any TMs that extend their inputs but I know my universal TM would fail because I haven’t added a way to extend the blank symbol. Also, there are a few corner cases where it would fail, like if the current state is larger than 25 bits, then it will just fail when it tries to write the state or if the encoded symbol you are writing is longer than 10 bits and the encoded symbol that its writing over is equal to or less than 10 bits. This leads me to believe that my universal turing machine does work on all or most turing machines with states less than 25 bits and encoded symbol writes to the read head less than 10 bits. I would also like to note that the reason I chose 10 bits is because I am assuming that its input is going to be a string encoded in ascii and that only goes up to 8 bits. The debug function I used was getnewConfigntimes which calls newConfigs n times and returns the nth config and every config before then.

The high level explanation of my universal TM is within my code and the transitions are commented. I start by writing the start state to where I am holding the current state, I then move the read head to point to the correct encoded symbol. I then check if the current state is in the list of final states, if it is then move to the final state, if it isn't then I reset everything that I have written over. I start comparing the read head to the transitions. I go through and compare each bit in the read head to the transitions and fail any transitions that don’t work. After this I have to go through the transitions and fail any that were supposed to fail but I didn’t catch. I then compare the current state to the list of transitions that haven’t failed yet and fail any of the ones that don’t match the current state. After, I go through the transitions again and fail any of the ones I missed that should have failed. At this point there should only be one transition that works, if there isn’t one then I go to a fail state and the TM fails. If there is one transition then I execute that transition by writing the symbol to the current spot on the read head, I fix the read head once I’m done writing it in case it was shorter than what was there before. I then write the new state to where I am holding the current state and fix it in case it is shorter again. Finally, I move the read head position and go back to the state that compares if the current state is in the list of final states and this just loops until I am in a final state or a fail state. Or it loops forever depending on the input TM.