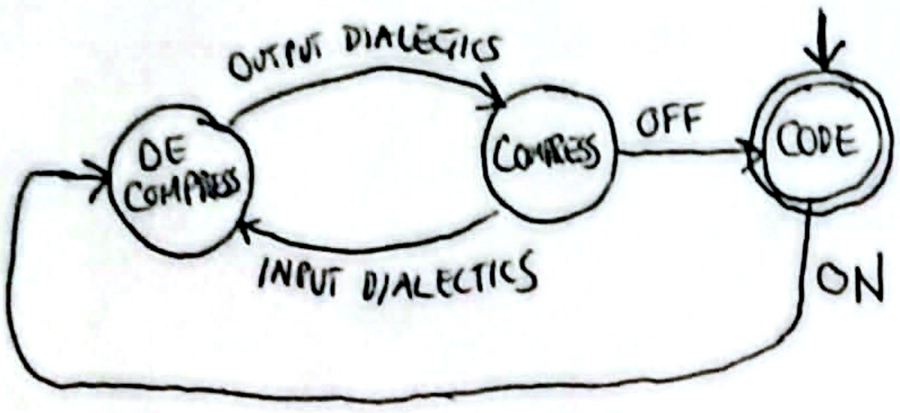
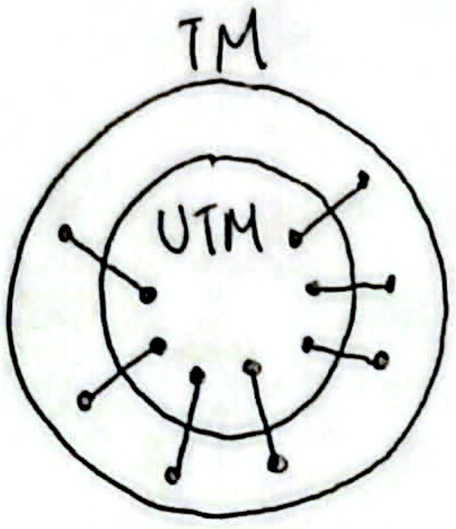
GIFFORD PROGRAMS



UNIVERSAL GIFFORD MACHINE DEFINITION



FOR EVERY TM, THERE IS A TM THAT IS COMPUTATIONALLY AS POWERFUL*, BUT LESS EFFICIENT. THIS NOTION OF EFFICIENCY IS DIFFERENT FROM EFFECTIVELY CALCULABLE ~~OR~~ AND/OR RECURSIVELY ENUMERABLE.

*COMPUTES THE SAME THING (SAME INPUT, SAME OUTPUT)

Q: WHAT MAKES A TM THAT COMPUTES THE SAME THING AS ANOTHER TM LESS EFFICIENT?

A:
1. USING MORE SYMBOLS (LARGER ALPHABET) THAN ANOTHER EQUIVALENT TM
2. NEEDING MORE STATES THAN ANOTHER EQUIVALENT TM
3. NEEDING MORE TRANSITIONS THAN ANOTHER EQUIVALENT TM
4. NEEDING A BLANK SYMBOL WHERE AN EQUIVALENT TM DOES NOT NEED TO READ/WRITE OR EVEN SPECIFY ONE, TO THE DEGREE THAT ANOTHER SYMBOL DOESN'T BECOME AN "EFFECTIVE BLANK" THAT EXACTLY PERFORMS THE BLANK SYMBOL'S FUNCTION.

Q: HOW IS ~~A~~ THE SET OF ALL TURING MACHINES *BIJECTED* WITH A UNIVERSAL TURING MACHINE?

A: REQUIRE THAT EACH TM IN {TM} IS IT'S MOST COMPUTATIONALLY EFFICIENT FORM FOR 1↔1 WITH {C}, THEN REQUIRE EACH UTM IN {UTM}* IS JUST A SLIGHTLY LESS EFFICIENT TM THAN EVERY OTHER ONE. THEN THE DISTINGUISHING BETWEEN WHETHER {TM} OR {UTM}* HAS THE "EFFICIENT" OR "INEFFICIENT" ~~FORMS~~ FORMS BECOMES MOOT