1 | Effective Computability

Dfn **1.1.** *Informal Definition: a function f is effectively computable if there are definite, explicit, mechanical instructions for computing each value of f.*

Rem 1.2. Turing computability is an attempt to make effective computability precise.

Prop **1.3.** *Turing Thesis: all effectively-computable functions are Turing-computable*

Rem 1.4. This proposition is not proven. But, there are many examples of effectively computable functions that are Turing computable, thus providing inductive support for the thesis

Dfn **1.5.** *Turing Machine consists of the following components: i) tape: infinitely long, marked into squares; endless in both directions; all but finitely many squares are blank at any stage.*
ii) a finite set $S_0, S_1, \ldots, S_n$. [BBJ: just S0 and S1] use S0 (or B or 0) for blank squares and 1 for S1 exactly one symbol is printed on each square
iii) machine is in one of finitely many internal states $q_1, \ldots, q_m$
iv) Actions at each step: machine scans the current square and reads the printed symbol conditional on the current symbol scanned and the current internal state, the machine performs one overt action and one covert action: a) Overt actions: (1) Halt the computation; (2) L: move one square left; (3) R: move one square right; (4) S0: write S0 in place of what is there; …; (n+4) Sn: write Sn in place of what is there. [BBJ: only S0 and S1.] b) Covert action: Assign a new internal state.

Rem 1.6. Graphical Representations of TMs

Rem 1.7. The numbering system (Arabic, Roman, etc) is irrelevant

Rem 1.8. Space, time limitations are irrelevant.

2 | Turing Computability

i) Monadic Notation. To represent one number a on a tape, we use a block of a 1's, with B everywhere else. To represent several numbers a1, a2, …, an, we use blocks separated by a single blank: a1 1's, B, a2 1's, B, …, B, an 1's. So to represent the triple (3, 5, 7), our tape would be: …BBB111B11111B1111111BBB…

Dfn **2.1.** *Turing-computable function: All such functions are defined using Turing machines that read and write only S0/B/0 and S1/1. a) Functions with one argument A Turing machine in internal state 1, scanning the leftmost 1 in a block of 1's on an otherwise blank tape, is said to be in standard starting position (s.s.p.); the same set-up (omitting internal state 1) is also standard final position (s.f.p.).*

3 | Examples of Turing Machines