basic recursion theory

specific models of computation delineate notions of

- algorithmic solvability
- computability (of partial functions)
- decidability (of sets or relations)
- recursive enumerability (of sets or relations)

different approaches
— Turing machines, register machines, μ-recursive functions, ...
lead to provably co-extensive notions

Church–Turing Thesis:
these models of computation capture the 'true' concepts of algorithmic solvability/decidability/enumerability

algorithms, recursion theory

general conventions:

work with finite alphabet $\mathbb{A}$ for coding of data (information)

$\mathbb{A}^*$: the set of $\mathbb{A}$-words $w \in \mathbb{A}^*$, of finite lengths $|w| \in \mathbb{N}$,
including the empty word $\Box \in \mathbb{A}^*$ of length $|\Box| = 0$

input (problem instances, arguments)
output (answers, function values)
} tuples over $\mathbb{A}^*$

key requirements for models of computation:

elementary steps of data manipulation,
simple small steps, local data access,
uniform rule-based control, ...

examples: Turing machines, register machines

register machines & programs

data format over alphabet $\mathbb{A} = \{a_1, \dots, a_r\}$:

registers $R_1, \dots, R_n$ for storing n-tuples $(w_1, \dots, w_n) \in (\mathbb{A}^*)^n$

elementary operations:

push and pop operations:
- $R_j := R_j + a_i$ (append letter a_i to content w_j of register R_j)
- $R_j := R_j - a_i$ (delete last letter in register R_j if w_j ends in a_i)

stop command:
- STOP (halt, terminate program execution)

control structure:

consecutively numbered program lines with conditional branching:

- IF $R_j = \Box$ THEN ℓ_0 ELSE ℓ_1 OR ... OR ℓ_r
  (case distinction on (last letter of) register content w_j of R_j)

R-programs & R-computation

execution of program P on input $\mathbf{w} \in (\mathbb{A}^*)^m$
proceeds by step-wise configuration updates,
and either

diverges: $\mathbf{w} \xrightarrow{\mathrm{P}} \infty$,
if STOP-command is never reached

or

terminates, halts: $\mathbf{w} \xrightarrow{\mathrm{P}} \mathrm{STOP}$ and $\mathbf{w} \xrightarrow{\mathrm{P}} \mathbf{w}'$
if STOP-command is reached (final configuration, output)

R-recursiveness, partial R-recursive functions

the R-program P on registers $R_1, \ldots, R_n$ over $\mathbb{A}$ computes the partial function:

$$\begin{array}{rcl} f\colon (\mathbb{A}^*)^n & \longrightarrow & (\mathbb{A}^*)^n \\ \mathbf{w} & \longmapsto & f(\mathbf{w}) \end{array}$$

where $\mathrm{dom}(f) = \{\mathbf{w} \in (\mathbb{A}^*)^n \colon \mathbf{w} \stackrel{\mathrm{P}}{\longrightarrow} \mathrm{STOP}\}$
and $\mathbf{w} \stackrel{\mathrm{P}}{\longrightarrow} f(\mathbf{w})$ for $\mathbf{w} \in \mathrm{dom}(f)$

suitable input/output conventions allow for computation of

- partial functions $f\colon (\mathbb{A}^*)^m \longrightarrow (\mathbb{A}^*)^{m'}$ for $m, m' \leqslant n$
- partial functions with boolean output in $\mathbb{B} = \{0, 1\}$

(R-)recursiveness, (R-)recursive enumerability

- a partial function is (R-)recursive/(R-)computable if it is the partial function computed by some R-program
- a relation $R \subseteq (\mathbb{A}^*)^n$ is (R-)recursive/(R-)decidable if its (total) characteristic function is computed by an R-program
- a relation $R \subseteq (\mathbb{A}^*)^n$ is (R-)recursively enumerable (r.e.) if it is the domain of some partial (R-)recursive function
- a relation $R \subseteq (\mathbb{A}^*)^n$ is (R-)co-r.e. if its complement is (R-)r.e.

remarks:

– unary $L \subseteq \mathbb{A}^*$ are called $\mathbb{A}$-languages

– recursive enumerability as semi-decidability:
R is recursive iff R is both r.e. and co-r.e.