HEURISTIC RULES FOR PROVING SOLVABILITY / UNSOLVABILITY

1. PROVE PROBLEM SOLVABLE (LANGUAGE RECURSIVE)

- invent a total Turing machine (it can be also multiple track or multiple tape or nondetermiministic TM), which solves the problem (or finite automata, pushdown automaton, regular expression, context-free or context-sensitive grammar). OR
- invent some Turing machine for both problem A and its complement $\overline{A}$ OR
- combine solution machine from total submachines OR
- reduce to some known solvable problem

2a. PROVE THAT PROBLEM IS AT LEAST PARTIALLY SOLVABLE (LANGUAGE REC. ENUMERABLE)

- invent some Turing machine (it can be also multiple track or multiple tape or nondeterministic TM), which solves problem (it doesn't have to halt always, in language recognition only "yes" -cases).
  → can be made from universal TM : simulate other machines to study their properties OR
- give unstricted grammar OR
- combine solution machine from some submachines OR
- reduce to some known partially solvable problem

2b. PROVE THAT PROBLEM IS ONLY PARTIALLY UNSOLVABLE (LANGUAGE RECURSIVE ENUMERABLE, BUT NOT RECURSIVE)

- show that its complement problem (language) is totally unsolvable (not recursive enumerable) OR

- reduce some known partially solvable problem (e.g. universal language U) to unknown one in question OR
- show that if solving machine is total, it causes contradiction :

COUNTER EXAMPLE METHOD

Is property P solvable?

1. suppose it is. Then we have total TM M_P, which solves it.
2. construct a new machine M_P', which performs P if the input machine doesn't.

   (diagram missing)
3. test M_P with code of M_P' as its input.
4. if contradiction, then such machine M_P cannot exist !

EXAMPLE: total halting tester machine M_H : cannot exist :

(diagram missing)
Now c_M' causes problem for M_H !

Task: what happens if property P is syntactic? e.g. "Machine M contains less than 10 states"?

3. PROVE UNSOLVABLE

- show that problem concerns some nontrivial semantic property of TM's. (Rice)
  → can be partially solvable
  → if complement is partially unsolvable, then it must be totally unsolvable (if it is sem. property) OR

- reduce some known totally unsolvable problem (e.g. diagonal language D) to unknown one OR
- show that its solvability (such TM) would cause contradiction

Meditate this:

UNSOLVABILITY OF A PROBLEM IS COMPUTATIONALLY UNSOLVABLE PROBLEM!