The fall-out: part 2

L_{halt} is *not* recursive but it *is* r.e.

Consider

$L_{\text{loop}} = \{\langle M \rangle \$x \mid M$ does *not* halt on input $x\}$,

i.e., complement of L_{halt} except for badly formed strings.

Question: Can this be recursive or r.e.?

Clearly: Can't be recursive.

THEOREM The complement of a recursive language is recursive.

THEOREM A language L is recursive if and only if both L and $\overline{L}$ are recursively enumerable.

COROLLARY The language L_{loop} is *not* recursively enumerable.

The uniform halting problem (reprise):

$L_{\text{uhalt}} = \{\langle M\rangle \mid M$ halts on all inputs $x \in \{0,1\}^*\}$.

We show this is not r.e.

• $\overline{L_{\text{uhalt}}}$ not r.e. either so need a subtler approach.

Reducing L_{loop} to L_{uhalt}

• We have: $\langle M\rangle\$x$ — instance of L_{loop}.

• We want: $\langle M_x\rangle$ — instance of L_{uhalt}.

Must satisfy:

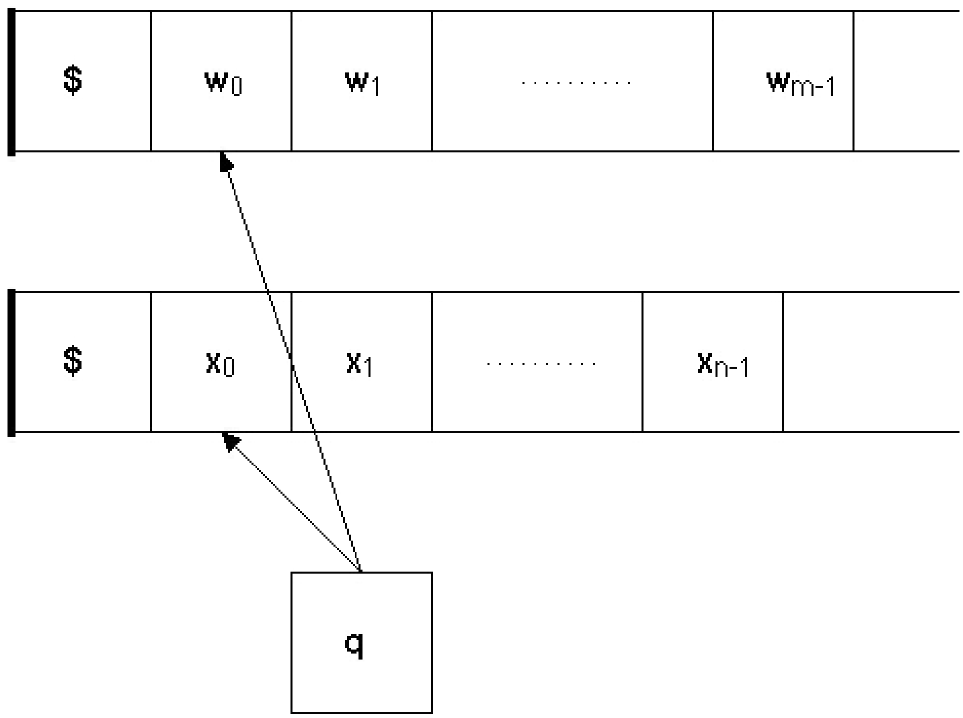
M does not halt on $x \Longleftrightarrow M_x$ halts on all inputs.

Key idea: M_x treats its input as a time bound for M when run with input x.

First build two tape TM M'_x with behaviour:

Given input $w \in \{ 0, 1 \}$

1. Mark leftmost squares;
2. Write x on tape 2;
3. Interpret w as a counter;
   Simulate M on x and decrement counter after each step;
4. When counter becomes 0;
   if simulation hasn't halted then halt
   else go into an infinite loop;



$
w0
w1
....
wm-1
$
x0
x1
....
xn-1
q

Proof systems for the uniform halting problem

Want: System to conduct formal proofs about the behaviour of TMs, i.e.,

- a formal language for making assertions about TMs,
- a collection of axioms and inference rules.

Make only two assumptions about this system:

(a) Language can express assertions of form 'machine M halts on all inputs', M an arbitrary binary TM.

(b) Proofs are machine checkable, i.e., there is a TM M_{check} that meets the specification:

INPUT: $\langle M \rangle \$ \pi$, i.e., binary Turing machine M, and $\pi \in \{0, 1\}^*$.

OUTPUT: 'Yes' if π encodes a valid proof of the assertion 'M halts on all inputs', 'no' otherwise.

Note: M_{check} always halts.

Obvious requirements on system:

- *Sound:* Proofs don't lie, i.e., provable $\rightarrow$ true.

- *Complete:* If M halts on all inputs then there is a proof, i.e., true $\rightarrow$ provable.

Fact: If such a proof system exists then L_{uhalt} is r.e.

Conclusion: No such system exists.