Recall: An editing system is a pair $\langle \Sigma, P \rangle$ where Σ is an alphabet and P is a finite set of pairs of the form (x,y) for $x,y \in \Sigma^*$.
Given strings $w, w' \in \Sigma^*$, say w edits to w' $(w \vdash w')$ if w' results from w by replacing a substring x in w with y, for some $(x,y) \in P$.
The editing problem (EP)
Instance: An editing system $E = \langle \Sigma, P \rangle$ and a string $w \in \Sigma^*$.
Question: Can w result in ε via a sequence of edits? [yes/no question]
Thm: EP is undecidable.
Proof: We m-reduce A_{TM} to EP $(A_{TM} \leq_m EP)$ via an m-reduction f defined as follows:
Given input $\langle M, w \rangle$, where $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej} \rangle$ is a TM and $w \in \Sigma^*$, we construct an instance $\langle E, \hat{w} \rangle$ of EP such that $\langle E, \hat{w} \rangle \in EP$ iff $\langle M, w \rangle \in A_{TM}$:
Let $E := \langle \hat{\Sigma}, P \rangle$ where $\hat{\Sigma} := \Gamma \cup Q \cup \{\$, \#\}$ where $\$, \#$ are distinct symbols not in $\Gamma \cup Q$.
And we build P as follows:
1. For every state $q \in Q$ and $a \in \Gamma$ such that $\delta(q,a) = (r,b,R)$ for some $r \in Q$ & $b \in \Gamma$, add the pair (qa, br) to P
2. For each $q \in Q$, $a \in \Gamma$ such that $\delta(q,a) = (r,b,L)$ (some r, b), add, for all $c \in \Gamma$, the pair (cqa, rcb) and $(\$qa, \$rb)$ to P
[strings to edit are the form $\$ID\#$ for IDs of M]
3. (Padding). Add $(\#, \sqcup\#)$ to P.
4. For every $a \in \Gamma$, add the pairs $(q_{acc}a, q_{acc})$ & (aq_{acc}, q_{acc}) to P.
5. Add $(\$q_{acc}\#, \varepsilon)$ to P.
End of the description of E.
Output $\langle E, \$q_0w\# \rangle$
[Talked thru the proof of correctness: idea is to allow edits corresponding to the successor relation of IDs of M, then wittle string down to ε if find an accepting ID.]

Resource-bounded computation (computational complexity theory).

Def: $f, g : \mathbb{N} \to \mathbb{R}^+$.

Say that $f(n) = O(g(n))$ [$f \in O(g)$] if $\exists C > 0$, $n_0 \in \mathbb{N}$, $\forall n \geq n_0$, $f(n) \leq C g(n)$.

Usually use n to mean the length of the input string. (default).

Let M be a TM and $t : \mathbb{N} \to \mathbb{R}^+$. Say that M runs in time $t(n)$ if for any input w

$\#\text{steps}(M \text{ on input } w) = O(t(n))$ where $n = |w|$.

[Assume by default that M is a decider.]

Def: $f, g : \mathbb{N} \to \mathbb{R}^+$. Write $f(n) = \underline{\text{Poly}}(g(n))$ to mean $f(n) = O(g(n)^k)$ for some constant k.

[Equiv, $\exists$ polynomial p such that $f(n) \leq p(g(n))$]

"f grows polynomially in g"

Note: $\text{Poly}(n) = \text{Poly}(n^2) = \text{Poly}(n^3) = \ldots$

Say that M runs in polynomial time (ptime) to mean $(\#\text{steps}(M \text{ on } w)) = \text{Poly}(|w|)$ for TM M & all inputs w.

Fact: ptime on a 1-tape TM = ptime on a multitape TM = ptime on a RAM = ptime on a pointer machine = ... (any reasonable model of computation)

Complexity Classes:

Def: P denotes the class of all decision problems (languages) that are decidable in ptime.

[deterministic ptime]

Ex:

Instance: A digraph G and vertices s, t of G.

Question: Does there exist a directed path from s to t?

[Graph Reachability Problem]

In P [use BFS, say]

Hamiltonian s-t path problem:

Instance: A digraph G and vertices s, t.

Question: Is there a path from s to t that runs through every vertex exactly once?

This problem is not known to be in P.

Def: NP ["nondeterministic ptime"] is the class of all languages L such that there exists a TM V such that on input $w\#y$, V halts in time polynomial in $|w|$ and $\forall w$,

$w \in L \Leftrightarrow \exists y, V(w\#y)$ accepts.

V is called a verifier

y is a candidate proof that $w \in L$. $V(w\#y)$ checks whether y is a legitimate proof.