undecidable: no algorithm

often, undecidable probs. are problems about code

Proving that a problem is undecidable:

1. proof by contradiction directly from TM

TM Y $\underset{\text{accept}}{\underline{\qquad\qquad}}$ <Y> <=> TM not $\underset{\text{accept}}{\underline{\qquad\qquad}}$ <Y>

2. Proof by reduction from a known undecidable prob.

today:

3. Rice's Theorem

Given <M>, does M accept $\underset{\text{e.g., } \varepsilon}{\underline{\qquad\qquad}}$?

<u>Rice's Theorem</u>

Let $\mathcal{J}$ be any set of languages so that:

(a) There is a TM Y such that

$ACCEPT(Y) \in \mathcal{F}$

(b) There is a TM N such that $ACCEPT(N) \notin \mathcal{F}$

Deciding whether $ACCEPT(M) \in \mathcal{F}$ is impossible.

Deciding whether a TM accepts languages w/ some property (e.g., contain ε) is undecidable, unless:

- No TM accepts languages w/ this property
- All TMs accept languages w/ this property

Problem: Given TM M, does M accept ε?

- $\mathcal{F}$ = all languages over Σ containing ε
- Y = a TM that accepts everything

$ACCEPT(Y) = \Sigma^* \in \mathcal{F}$ because $\varepsilon \in \Sigma^*$ ✓

- N = a TM that accepts nothing

$ACCEPT(N) = \emptyset \notin \mathcal{F}$ because $\emptyset \notin \mathcal{F}$ ✓

By Rice's Thm, deciding whether $ACCEPT(M) \in f$ is impossible.

$ACCEPT_\varepsilon = \{\langle M\rangle : M \text{ accepts } \varepsilon\}$ is undecidable.

Some scratch work:

is $\{\varepsilon, 0, 01\} \in f$

is $11^* \in f$? no

$\{1, 11, 111, \dots\}$

$f = \{\{1, 10, \varepsilon\}, \{\varepsilon, 1, 11, 111, \dots\}\}$

f = a set of sets

Σ^* = a set of strings

1. GRIZ = { ⟨M⟩ : M accepts the string grizzlies }

Griz is undecidable.

Proof

Let $\mathcal{J}$ = the set of all langs containing the string grizzlies

Let Y = the TM accepting all strings

ACCEPT(Y) $= \Sigma^* \in \mathcal{J}$

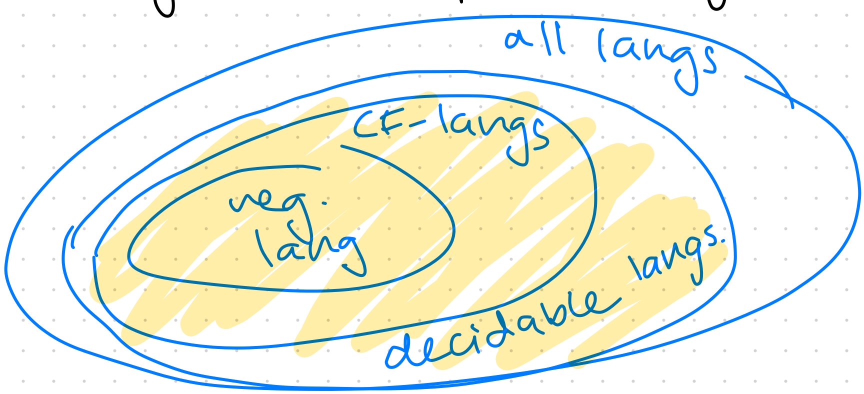
Let N = the TM accepting nothing

ACCEPT(N) $= \emptyset \notin \mathcal{J}$

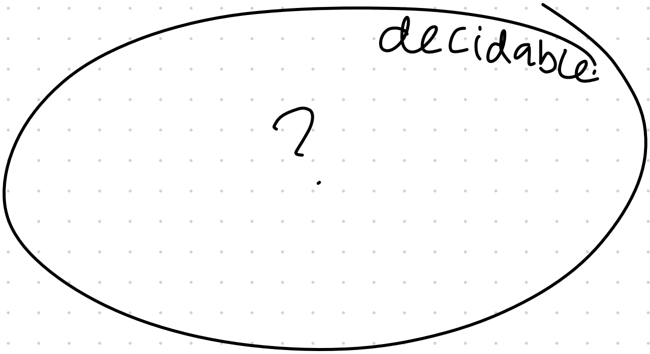
By Rice's Thm, deciding whether ACCEPT(M) $\in \mathcal{J}$ is impossible

---

Summary of computability:



how fast?

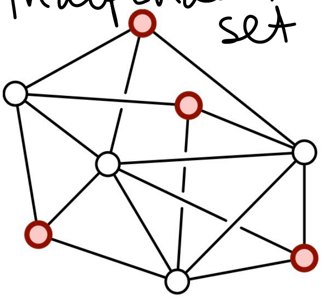


our tool for "how fast?": reductions

undecidability: suppose A is decidable.
Then B is too.

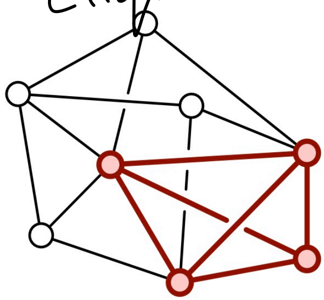
runtime: suppose A is solvable quickly.
Then B is too.

independent set



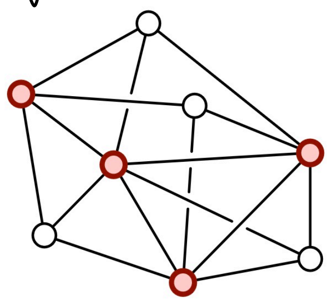
Q: largest

clique



Q: largest

vertex cover



Q: smallest

Suppose I have magic black box to solve MaxClique

MaxIndSet can now be solved.

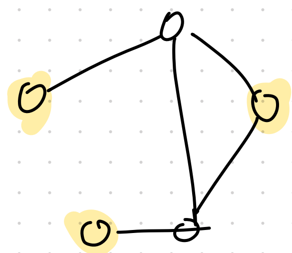
Alg for MaxIndSet(G):

build a new graph $\bar{G}$ with:

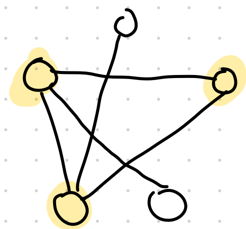
$\bar{V} = V$ $|V|$ time

$\bar{E} = \{uv : uv \notin G\}$ $O(|V|^2)$ time

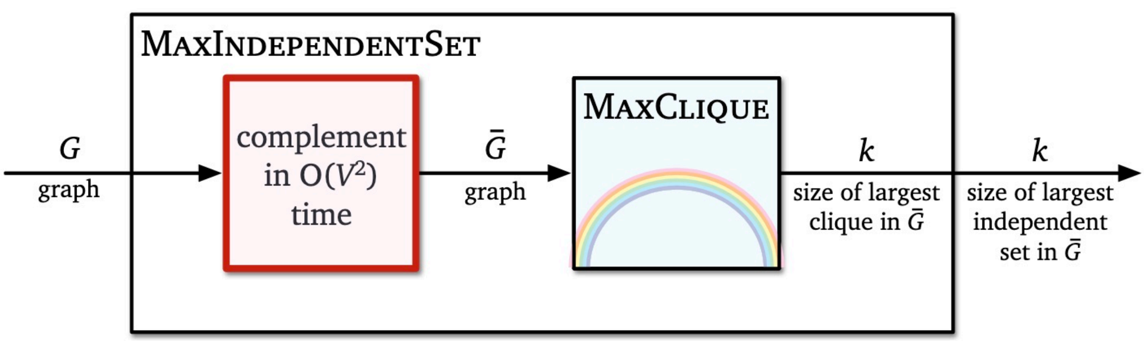
Solve MaxClique on $\bar{G}$ time to solve MaxClique



G



$\bar{G}$



Max Ind Set reduces to Min Vertex Cover

Assume I have a black box solving MinVC in polynomial time.

MaxIndSet(G):
    Let n = # of verts in G
    return n - MinVertexCover(G)