Interactive proof system

→ Traditional mathematical proofs are static objects

→ ; A prover $\underline{P}$ write down a sequence of mathematical statements, and then at some later time a verifier $\underline{V}$ checks that these statements are consistent and correct.

→ Over the years, computer science has changed the notion of a mathematical proof.

→ First such change was the observation that for all practical purposes the verification procedure should be efficient

i.e $\underline{V}$ should not have expend large amount of efforts to verify the proof of a claim.

(Much less than $\underline{P}$ expended to find the proof).

→ This notion of "efficient verification" corresponds to the complexity class NP

Defn: A language L belongs to NP, iff $\exists$ an efficient algorithm $\underline{V}$ such that the following conditions hold

Completeness: $\forall x \in L$, $\exists$ a proof π that makes $\underline{V}$ accept $\Rightarrow V(x, \pi) = 1$.

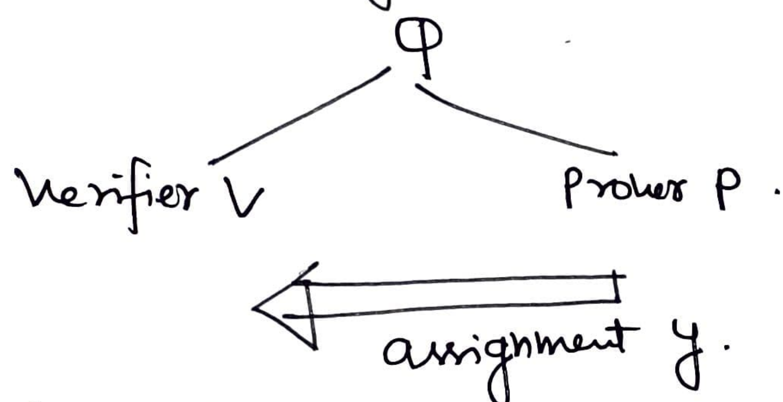
Soundness: $\forall x \notin L$, for all claimed proof π^*, V rejects: $V(x, \pi^*) = 0$

Simplified form:

NP as a proof system

- if $L \in NP$, we can think of
- a polynomial-time verifier $\underline{\underline{V}}$ and.
- an all powerful prover $\underline{\underline{P}}$
- They are both given input w
- $\underline{\underline{P}}$ needs to convince $\underline{\underline{V}}$ that $w \in L$.

Example: proof system for SAT.



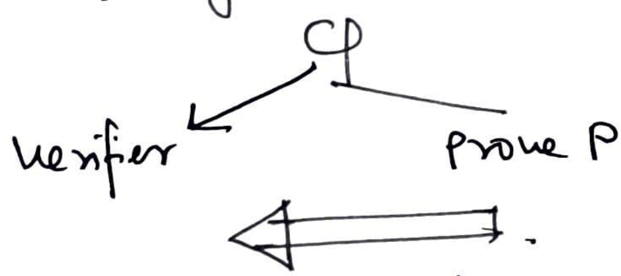
V accepts
if y satisfies φ.

if $\varphi \notin SAT$, then no P makes V accept.
whatever P sends, V will not accept.

Even though the verification procedure is now efficient, the proof is still a static object. Computer scientists in the 80's and 90's changed this view by introducing interaction and randomness into the mix.

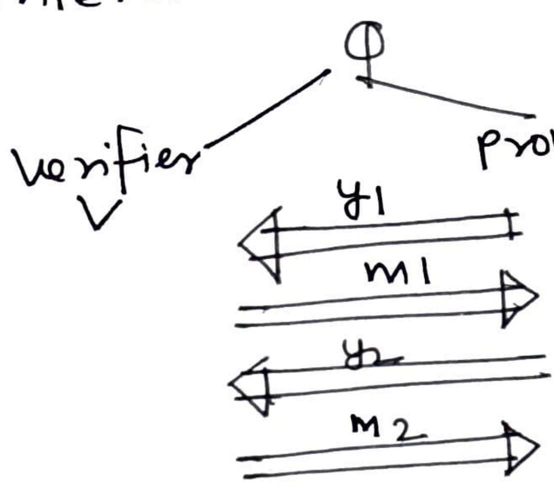
→ The prover and verifier were no longer required to be deterministic, and can now talk to each other.

Ex: proof system for not SAT.



can a prover send some y that convinces V that φ is not satisfiable.
(Believed to be impossible)

fact: $\exists$ proof system for not SAT with interaction and randomization



V accepts with high probability $\iff \varphi \notin SAT$.

Defn: A language L has an interactive proof (and belong to the class IP) if there exists an efficient randomized interactive algorithm $\underline{V}$ that satisfies the following conditions.

Completeness:

$\forall x \in L$, there exists an unbounded interactive 'prover' algorithm $\underline{P}$ such that $\underline{V}$ interacts with $\underline{P}$ and accepts with high probability

$$\Pr[\underbrace{\langle \underline{P}, \underline{V} \rangle}_{\text{Interaction in b/w P and V}}(x) = 1] \geq 2/3 .$$

Soundness:

$\forall x \notin L$, $\forall$ algorithms P^*, $\underline{V}$ interacts with P^* and rejects with high probability

$$\Pr[\langle P^*, \underline{V} \rangle (x) = 1] \leq 1/3 .$$

Note: See my other slides for simpler definitions of Soundness and completeness.