Non-Approximability Results
(2nd part)

Summary

- The PCP theorem
  - Application: Non-approximability of MAXIMUM 3-SAT

Non deterministic TM

- A TM where it is possible to associate more than one next configurations to the current one.
- Given an input, more than one computations are possible.
- A string is *accepted* if at least one such computation halts in accepting state
- A string is *rejected* if all computations halts in rejecting state

Oracle TM

- An oracle TM has
  1. an associated oracle languages A
  2. an oracle tape
  3. Three new states: q_Q, q_Y, q_N
- Any time it enters in state q_Q, the next step it enters in state q_Y if the current string in the oracle tape is in A, q_N otherwise
- The query costs 1 step

Oracle TM

- Changing *A* may imply that also the language recognised changes
- *For any complexity class C and any language A, let C^A be the set of languages recognised with complexity C by an oracle TM with oracle language A.*
  - Turing reducibility is given in terms of oracle TM
  - A common representation: $NP \subseteq P^{SAT}$

Probabilistic TM

- An probabilistic TM has
  1. A read only tape: *random tape*
  2. A new states: q_r.
- Any time it enters in state q_r, the next step it enters in a state in according to the current symbol of random tape and advances the random tape head by one cell to the right.
- For any input, the computation depends on the random string initially contained in the random tape

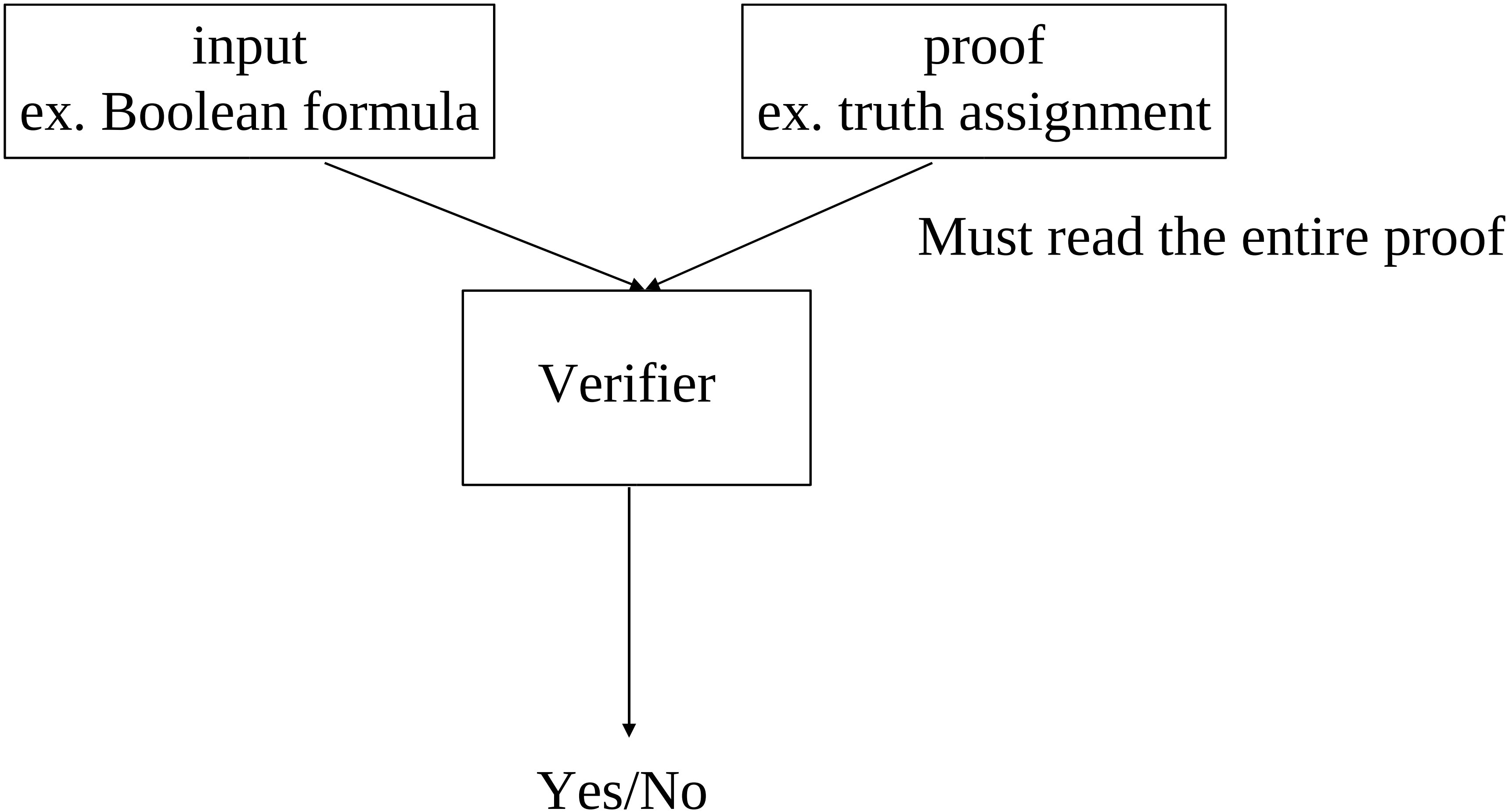
Probabilistic TM

- A PTM is *r(n)-restricted* if, for any input *x* of length *n*, it enters at most *r(n)* times in state q_r.
- Class RP (Random polynomial) = { L | there exists a polynomial-time PTM such that for any input *x*,
  - If $x \in L$, then *x* is accepted with probability $\geq$ ½
  - If $x \notin L$, then *x* is rejected with probability 1
  - r(n) has to be polynomial in $|x|$ at most.

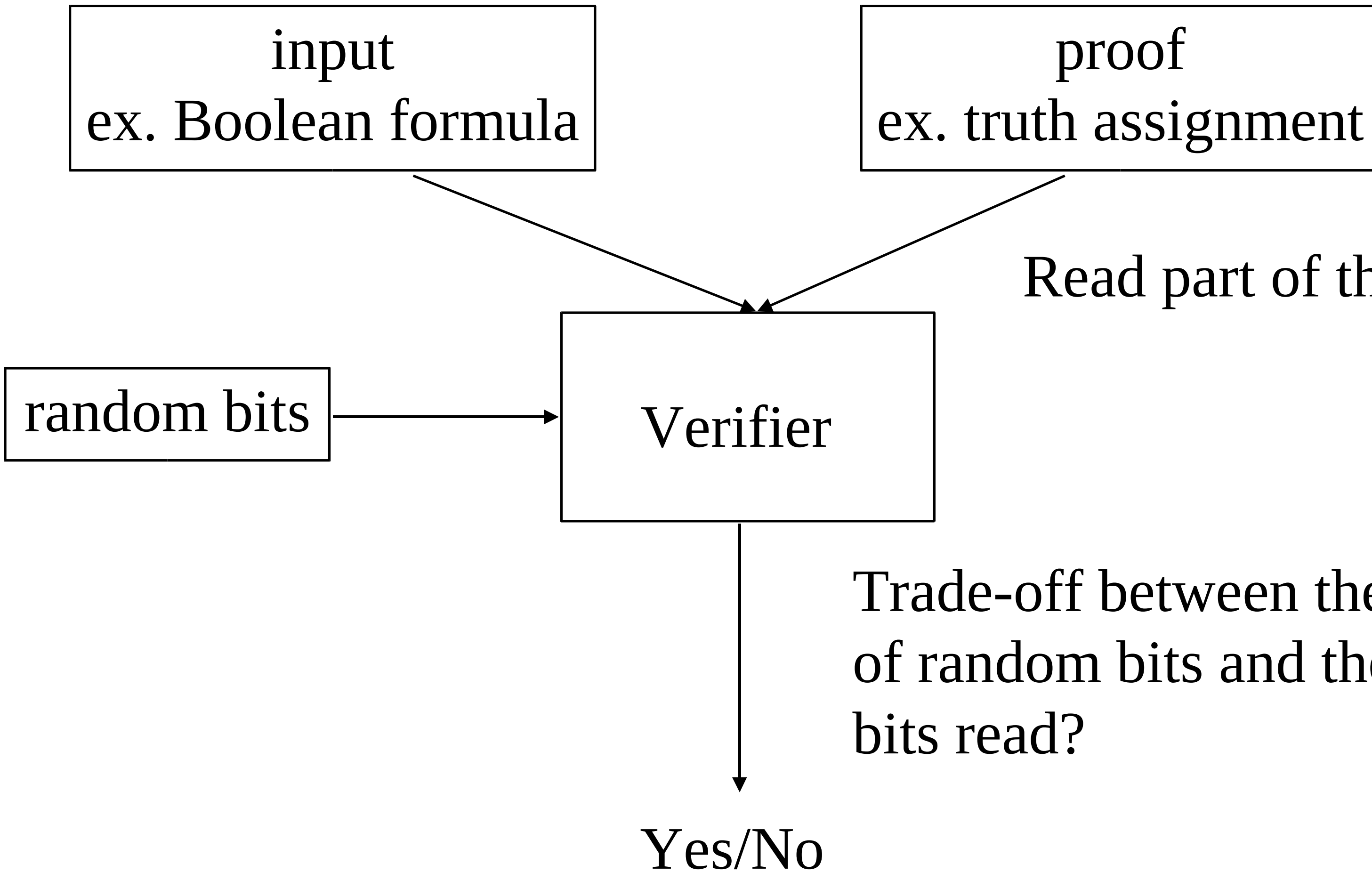
Verifier

- A *verifier* is a polynomial-time oracle probabilistic TM which uses the oracle to access a proof on a random access basis: when the oracle is given a position (address), it returns the value of the corresponding bit
  - Given a proof π, the corresponding oracle language X_π is the set of addresses corresponding to 1-bits
- The computation consists of 2 phases:
  1. The verifier uses the random tape to determine which bits in the proof will be probed.
  2. The verifier deterministically reads these bits and, finally, accepts or rejects depending on their values.

Deterministically checkable proofs



Probabilistically checkable proofs



Read part of the proof

Trade-off between the number of random bits and the number of bits read?

PCP[r,q]

- A decision problem P belongs to PCP[r,q] if it admits a polynomial-time verifier A such that:
  - For any input of length n, A uses $r(n)$ random bits
  - For any input of length n, A queries $q(n)$ bits of the proof
  - For any YES-instance x, there exists a proof such that A answers Yes with probability 1
  - For any NO-instance x, for any proof A answers Yes with probability less than ½

(the probability is taken over all random binary strings of length r(|x|), uniformly chosen)

The PCP theorem

- Given a class F of functions, PCP(r, F) is the union of PCP[r,q], for all $q \in \mathsf{F}$
- By definition, NP=PCP(0, poly) where poly is the set of polynomials

Theorem: NP=PCP(O(log),O(1))

- Proving that NP includes PCP(O(log), O(1)) is *easy*
- Proving that NP is included in PCP(O(log), O(1)) is hard (complete proof is more than 50 pages, involving sophisticated techniques from the theory of error-correcting code and the algebra of polynomials in finite fields)

Inapproximability of satisfiability

- Gap technique
  - Intuitive motivation: gap in acceptance probability corresponds to gap in measure
- Reduction f from SAT such that
  - If x is satisfiable, $m^*(f(x))=c(x)$ where $c(x)$ is the number of clauses in $f(x)$
  - If x is not satisfiable, $m^*(f(x))<c(x)/(1+\mathrm{g})$ with $g>0$
  - Gap: g not explicitly computed
- **Theorem:** MAXIMUM SAT is not polynomial-time r-approximable for $r<1+g$

Inapproximability of satisfiability

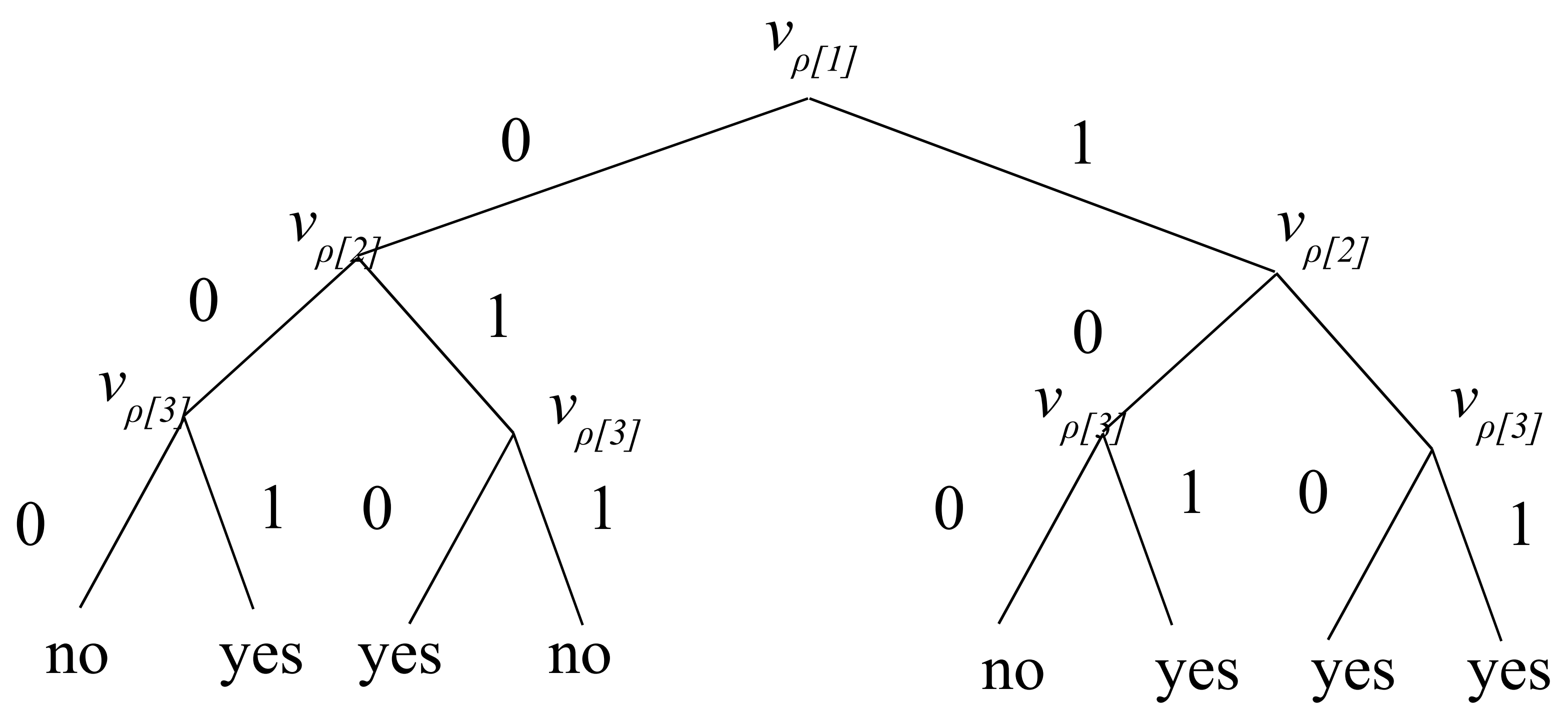
- The proof is shown for MAX 3-SAT problem
  - Let L the 3-SAT NP-complete problem
  - There exists a polynomial-time ($r(n)$, q) verifier for L
    - $r(n) = O(\log n)$, n = dimension of φ
    - q is constant. We assume $q > 2$
    - w.l.o.g., we assume that the verifier asks exactly q bits of the proof
  - Given x, we will construct in polynomial-time an instance C of MAX 3-SAT s.t. if $x \in L$, then C is satisfiable, otherwise there is a constant $\varepsilon > 0$ such that at least a fraction ε of clauses in C cannot be satisfied.

Inapproximability of satisfiability

- Consider a possible proof string π
  - For each bit of π we introduce a boolean variable
  - We do not need to consider proofs longer than $q2^{r(n)}$
  - $r(n) \leq$ c log n, for some constant c
  - The number of new boolean variables is bounded by $q\ n^{c}$
  - *v-th* variable stands for the statement "*the v-th bit in π is 1*"
- Consider a possible random string ρ
  - Let $v_{\rho[1]}, v_{\rho[2]}, v_{\rho[3]}, \ldots, v_{\rho[q]}$, be the q variables that correspond to the q bits that verifier will read given the random string ρ

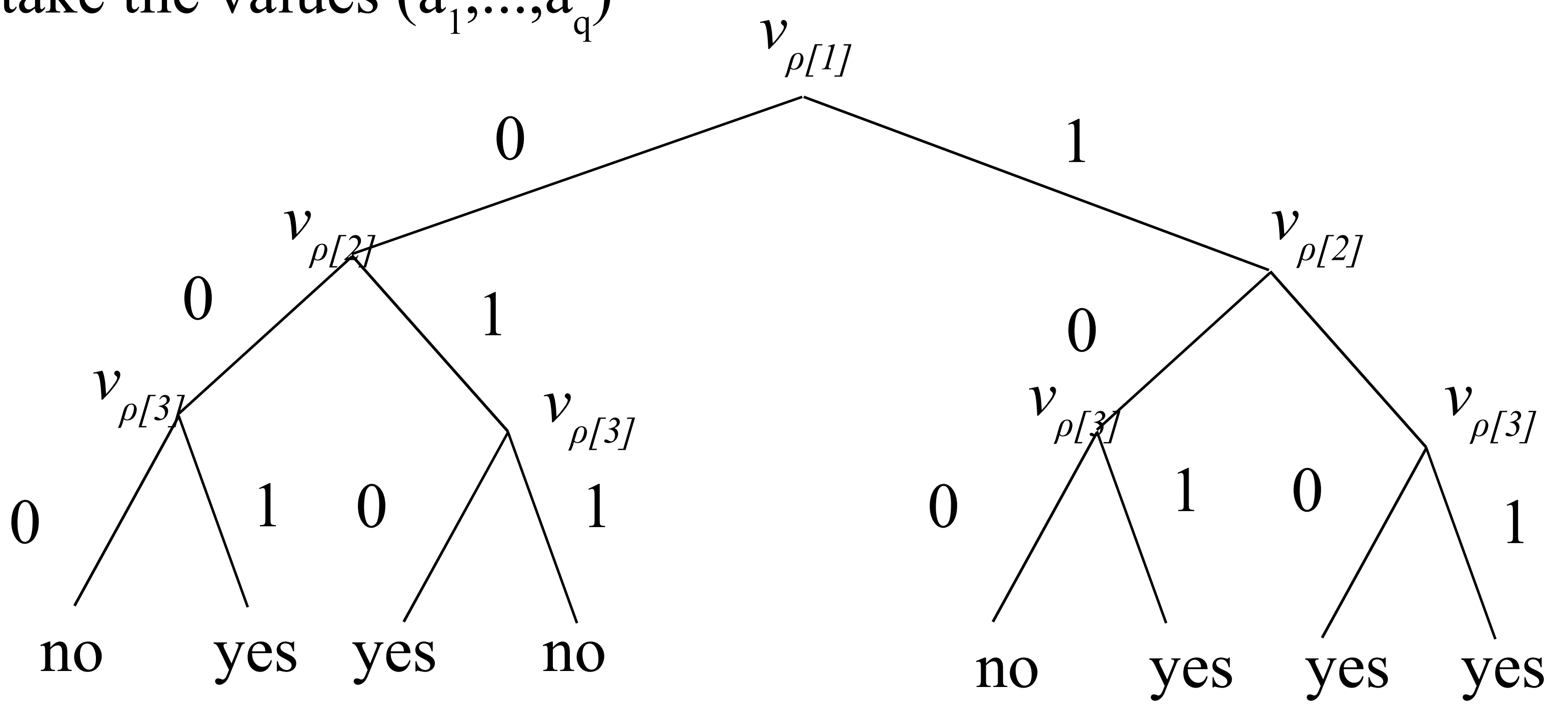
Inapproximability of satisfiability

- In general, for some q-tuples of values, the verifier will accept and, for some other tuples, it will reject.



Inapproximability of satisfiability

- Let A_ρ be the set of q-tuples for which the verifier rejects. For each tuple $(a_1,\ldots,a_q) \in A_\rho$, build a clause of q literals which is true iff the prof bits do no take the values $(a_1,\ldots,a_q)$



$(v_{\rho[1]} \vee v_{\rho[2]} \vee v_{\rho[3]}), (v_{\rho[1]} \vee \neg v_{\rho[2]} \vee \neg v_{\rho[3]}), (\neg \boldsymbol{v_{\rho[1]}} \vee v_{\rho[2]} \vee \boldsymbol{v_{\rho[3]}}), (\neg v_{\rho[1]} \vee \neg v_{\rho[2]} \vee \neg v_{\rho[3]})$

Inapproximability of satisfiability

- Let C_R the set of this clauses. $|C_R| \leq 2^q\, 2^{r(n)} \leq 2^q\, n^c$
- If x is in L, there exists a proof $\pi(x)$ such that the verifier will accept for all random strings ρ. If we set the variables as $\pi(x)$ shows, then every clause in C_R will be satisfied
- If x is not in L, we know that regardless of the proof π, there will be at least $2^{r(n)}/2$ of the random string ρ for which the verifier will reject. Hence,

$$(2^{r(n)}/2)/(q-2)2^q\, 2^{r(n)} \leq (2^{r(n)}/2)/2^q\, 2^{r(n)} = 2^{-(q+1)}$$

clauses are unstatisfiable, the gap!

The NPO world if P≠NP

NPO	MINIMUM TSP
APX	MINIMUM BIN PACKING MAXIMUM SAT MINIMUM VERTEX COVER(⇩ ?) MAXIMUM CUT(⇩ ?)
PTAS	MINIMUM PARTITION
PO	MINIMUM PATH

MINIMUM GRAPH COLORING? Certainly not in PTAS