P, NP, PSPACE, etc.

Running Time of a Turing machine (TM, aka algorithm)
- for input x: number of steps s(x)
- for n-bit input: t(n) := max{x(x) : x ∈ L, |x| ≤ n} (worst-case)
- TM is poly-time if t(n) is a polynomial function

Complexity Classes
- P = {L : ∃ poly-time TM that decides L}
- NP = {L : ∃ poly-time comp. function ϕ : {0, 1}∗×{0, 1}∗→{0, 1} s.t. x ∈ L ⇔ ∃w (ϕ(x, w) = 1 ∧ |w| ≤ poly(|x|))}
  (also: NP = {L : ∃ non-det. poly-time TM that accepts L})
- NP-hard = {L : ∀L' ∈ NP: L' can be reduced to L}
- NP-Complete = NP ∩ NP-hard
- PSPACE = {L : ∃TM that accepts L with poly memory (in any time)}

P, NP, PSPACE, etc.

Polynomial, Negligible, Noticeable

Function f : N → R
- f is polynomial ⇔ ∃c ∃n0∀n ≥ n0 : f(n) ≤ nc
- f is negligible ⇔ ∀c ∃n0 ∀n ≥ n0 : f(n) ≤ 1/nc
- f is noticeable ⇔ ∃c ∀n0 ∀n ≥ n0 : f(n) ≥ 1/nc
- f is overwhelming ⇔ 1 − f is negligible

Implications
- poly × poly = poly; poly(poly) = poly
- poly × noticeable ⊆ negligible
- (poly × noticeable) ∩ overwhelming = {}

Interactive Proofs of Statements

Def: An interactive proof for language L is a pair (P, V) of int. programs s.t.
- i) ∀x : running time of V is polynomial in |x|
- iii) ∀x ∈ L, ∀V' : Pr((P, V') accepts) ≤ 1/2 [q = 1/2]

Examples: Sudoku, GI, GNI, Fiat-Shamir.

Remarks
- Constants p, q are arbitrary, could be p = 1 − 2−|x| and q = 2−|x|
- However: only NP-languages have proofs with q = 0
- If iii) holds only for poly-time P: interactive argument (not a proof)
- Probabilistic P are not more powerful than deterministic P

Def: IP = set of L which have an interactive proof.

Theorem: IP = PSPACE.

Zero-Knowledge

Idea: Protocol (P, V) has transcript T, simulator S outputs similar T'.

Def: (P, V) is zero-knowledge (ZK) ⇔ ∀ poly-time V' ∃ S:
- i) Transcript T of (P ↔ V) and output T' of S are indistinguishable.
- ii) Running time of S is polynomially bounded.

Def: (P, V) is black-box zero-knowledge (BB-ZK) ⇔ ∃ S ∀ V:
- i) Transcript T of (P ↔ V) and output T' of S in (S ↔ V') are indisting..
- ii) Running time of S is polynomially bounded.

Def: (P, V) is honest-verifier zero-knowledge (HVZK) if S exists for V' = V.

c-Simulatability and Zero-Knowledge

Definition: A three-move protocol (round) with challenge space C is c-simulatable if for any value c ∈ C one can efficiently generate a triple (t, c, r) with the same distribution as occurring in the protocol (conditioned on the challenge being c), i.e., the conditional distribution P_{TRC} is efficiently samplable.

Lemma: A 3-move c-simulatable protocol is HVZK.
  (Assumption: challenge is efficiently samplable)

Lemma: A HVZK round with c uniform from C for poly-bounded |C| is ZK.

Lemma: A sequence of ZK protocols is a ZK protocol.

Theorem: A protocol consisting of c-simulatable rounds, with uniform challenge from a (per-round) polynomially bounded space C', is perfect ZK.