Contents

1 Course introduction

Professors: Geoffroy Couteau (couteau@irif.fr) & Alain Passelègue (alain.passelegue@ens-lyon.fr)

First part = complexity theory. End goal = interactive proofs. Second part = cryptography applications of complexity theory.

Notation 1.
- $x \xleftarrow{\$} A$ *means we assign* x *a random value from set* A
- $\{0,1\}^n$ *denotes the set of bit strings of length* n
- $\{0,1\}^*$ *denotes the set of all bit strings*
- $|x|$ *denotes the length of a bit string* x
- $\mathrm{O(g)}$ *and* $\Omega\mathrm{(g)}$ *are the standard Landau notations*

Definition 1. *P runs in time* $\mathrm{poly}(n) \Leftrightarrow \exists$ *polynomial* Q*, runtime of P bounded by* $\mathrm{Q}(n)$

Definition 2 (Search problem)**.** $\mathrm{R} : \{0,1\}^* \times \{0,1\}^*, \mathrm{R}(x) = \{y : \exists y, (x,y) \in \mathrm{R}\}$ f *solves the problem* R *if* $\forall x, \mathrm{f}(x) \in \mathrm{R}(x)$ *with* $\mathrm{f} : \{0,1\}^* \to \{0,1\}^* \times \bot$

Definition 3 (Decision problem)**.** *Let* $S \subset \{0,1\}^*$. $\mathrm{f} : \{0,1\}^* \to \{0,1\}$ *solves the problem* S *if* $S = \{x : \mathrm{f}(x) = 1\}$

Definition 4 (Language)**.** *A language is a subset of* $\{0,1\}^*$ $\mathcal{L}_n = \mathcal{L} \cap \{0,1\}^n$

Definition 5 (Turing machine)**.**
- *Environment = infinite band*
- *We can write things on the tape with an alphabet* Σ *(usually* $\{0, 1, blank\}$*)*
- *We have a pointer indicating where we are on the tape*
- *A set of states* Q*, including an initial state* q_0 *and a subset of final states* Q_{halt}
- *transition function:* $\mathrm{T} : \Sigma \times Q \to \Sigma \times Q \times \{-1, 0, +1\}$

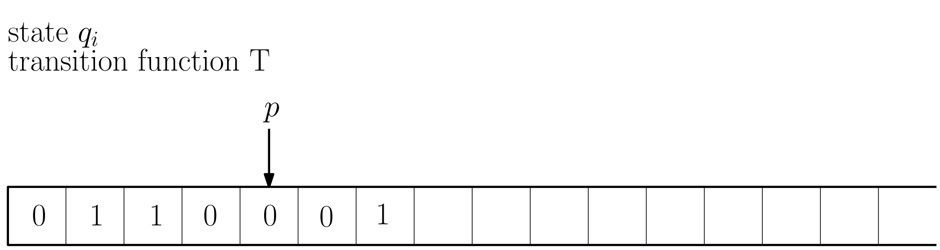

Figure 1: A Turing machine is an infinite tape, alongside some rules to work on this tape

Definition 6. *M* computes f *if $\forall x \in \{0,1\}^*$, when M is started with x on its tape, it halts after a finite number of steps with* $\mathrm{f}(x)$ *written on its tape.*

Definition 7 (Probabilistic Turing machine)**.** *Turing Machine with two tapes:*

- *random tape = filled randomly with* $0s$ *and* $1s$
- *work tape*

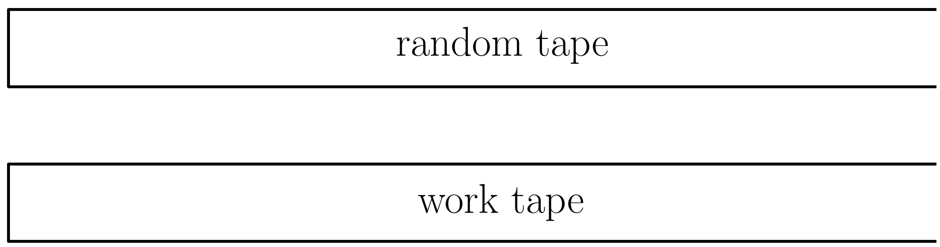

Figure 2: A probabilistic Turing machine

Notation 2. $\mathrm{M}(x,r)$ *means* M *is executed on input x with random tape r.*

Definition 8 (Interactive Turing machine)**.** *Five total tapes:*

- *input tape*
- *random tape*
- *work tape*
- *communication tape 1: read only*
- *communication tape 2: write only*

Definition 9 (Communication between two Turing machines)**.** *Two machines* M_1 *and* M_2 *are said to be communicating if they share their communication tapes: one comm. tape 1 for one is comm. tape 2 for the other.*

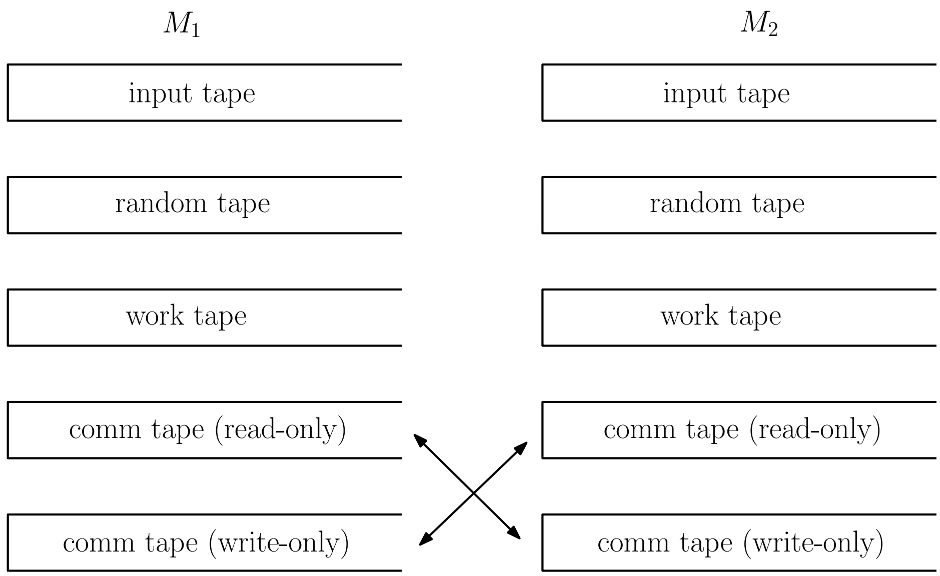

Figure 3: Two communicating machines: the arrows mean the tapes are the same

Definition 10 (Oracle Turing machine)**.** M *has oracle access to* f *if in addition to its usual behavior,* M *can query a value* x *to* f *in which case* $\mathrm{f}(x)$ *is written on the tape.*

Notation 3.
- *PPT: probabilistic polynomial time*
- *PT = polynomial time*

A PPT Turing machine M is a probabilistic machine that always halts after some polynomial number of steps. $\exists$ PPT TM M $\Leftrightarrow$ $\exists$ probabilistic TM M and a polynomial P such that on any input $x \in \{0,1\}^*$, M halts after at most $\mathrm{P}(|x|)$ steps.

Definition 11 (Complexity)**.** *P = languages $\mathcal{L}$ which can be decided by a PT* M *NP = languages $\mathcal{L}$ if there exists a PT relation* $\mathrm{R} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ *such that* $\mathcal{L} = \{x : \exists y, |y| = \mathrm{poly}(x) \wedge \mathrm{R}(x,y) = 1\}$

Definition 12 (Proof system)**.** *$\mathcal{L} \in NP$ has a proof system as it has the following properties:*

1. *completeness: $\forall x \in \mathcal{L}$, there exists a short and efficiently verifiable proof of $x \in \mathcal{L}$*
2. *soundness: no $x \notin \mathcal{L}$ has a short accepting proof*

Definition 13 (NP-hardness and NP-completeness)**.** *$\mathcal{L}$ is NP-hard if $\forall \mathcal{L}' \in NP$, $\mathcal{L}' \leq_P \mathcal{L}$ ($\mathcal{L}'$ can be reduced to $\mathcal{L}$ in polynomial time). This means that there exists a PT machine* M *such that* $\forall x \in \{0,1\}^*, x \in \mathcal{L}' \Leftrightarrow \mathrm{M}(x) \in \mathcal{L}$*. Such a reduction is also called a Karp-reduction.*

We say $\mathcal{L}$ is NP-complete if it is NP-hard and $\mathcal{L} \in NP$.

Definition 14 (Another definition of reduction)**.** *If* f *decides $\mathcal{L}$ then* M^{f} *decides $\mathcal{L}'$. This is called a Cook-reduction or a Turing-reduction. This definition is more general, but both definitions are equivalent for decision problems.*

Example 1 (Some NP-complete problems)**.**
- *SAT: is this CNF satisfiable? CNF =* $\bigwedge(\bigvee$ *literals*$)$
  proof by Cook in 1971 and Levin in 1973

- 3-*SAT: SAT with only* 3 *literals per clause* ($\bigvee$) ***exercise: polynomial reduction from SAT to 3-SAT***

- *Graph Hamiltonicity: is there a hamiltonian cycle in the graph? (cycle that goes through each node exactly once)*

- 3-*coloring: coloring a graph with* 3 *colors*

Theorem 1 (Ladner's theorem). $P \neq NP \Rightarrow \exists$ *"NP-intermediate" problems* $= NP \backslash NP$*-hard*

Simplified proof. Let $\mathcal{L} \in NP \backslash P$ such that the best algorithm deciding $\mathcal{L}$ runs in time $n^{\log(n)}$. Define $\mathcal{L}' = \{(x, y) | x \in \mathcal{L} \wedge |x| + |y| = |x|^{\log(\log(|x|))}\}$ $\mathcal{L}' \in P$? If $\mathcal{L}' \in P$, then $\mathcal{L}$ can be decided in time poly$(|x|^{\log(\log(|x|))})$ which is a contradiction. $\mathcal{L}'$ NP-complete? If so, we can reduce $\mathcal{L}$ to $\mathcal{L}'$ in polynomial time. Let $N = |x|, N^{\log(\log(N))} = n$. $\mathcal{L}'$ is decidable in time $N^{\log(N)}$ which implies $\mathcal{L}$ is solvable in time poly$(N^{\log(N)})$ and $N^{\log(N)} = n^{o(\log(n^{o(1)}))}$ which is a contradiction.

□

Definition 15 ($coNP$). $\mathcal{L} \in coNP$ *if* $\mathcal{L}^c \in NP$

Example 2. *UNSAT is complete for coNP*
TAUTOLOGY is complete for coNP

Remark 1. *NP:* $x | \exists y, \mathrm{R}(x, y) = 1$ *coNP:* $x | \forall y, \mathrm{R}(x, y) = 1$

Definition 16. $\forall k \in \mathbb{N}, \Sigma_k$ *is the class of* $\mathcal{L}$ *such that there exists a polynomial* P *and PT TM* M *such that* $x \in \mathcal{L}$ *iff* $\exists y_1 \in \{0, 1\}^{\mathrm{P}(|x|)}, \forall y_2 \in ..., \exists y_3 \in ..., ..., \mathrm{M}(x, y_1, ..., y_k) = 1$

$$PH = \bigcup_{k \in \mathbb{N}} \Sigma_k$$

Remark 2. $\Sigma_0 = P, \Sigma_1 = NP$

Theorem 2. $P = NP \Leftrightarrow P = PH$

Proof. $\mathcal{L} \in \Sigma_{k+1}$: $\mathcal{L} = \{x : \exists y', (x, y') \in \mathcal{L}'\}$ where $\mathcal{L}' \in \Pi_k$

$\mathcal{L} \in \Sigma_2 \Leftrightarrow \exists$ polynomial P, $\mathcal{L}' \in coNP : \mathcal{L} = \{x : \exists y, (x, y) \in \mathcal{L}'\}$ If $P = NP$ then $P = coNP$ then $\Sigma_2 = NP = P$ with the above equivalence.

□

Definition 17 ($PSPACE$). $\mathcal{L} \in PSPACE(\mathrm{S}(n))$ *if there exists a TM* M *that decides* $\mathcal{L}$ *and the number of cells that will be non-blank at any time during the computation is bounded by* $\mathrm{S}(|x|)$

$$PSPACE = \cup_{c>0} SPACE(n^c)$$

Example 3. $TQBF : \psi = Q_1 x_1 Q_2 x_2 ... Q_n x_n \phi(x_1, ..., x_n)$ *where* $Q_i \in \{\forall \exists\}, x_1, ..., x_n$ *are bits,* $|\phi| = m$ *TQBF is PSPACE-complete* ***exercise:*** *TQBF* ***is in*** *PSPACE*