Algorithms and Complexity

Spring 2018

Aaram Yun

This page is intentionally left blank

Today

- A few classes of problems
  - $\mathcal{PF}, \mathcal{PC}, \mathcal{P}, \mathcal{NP}$

Polynomially-bounded relations

x y

- $R \subseteq \{0,1\}^* \times \{0,1\}^*$; a search problem
- We want to talk about efficiently solvable search problems
- One prerequisite: polynomial boundedness
  - R is *polynomially bounded*, if there exists a polynomial $p()$ such that, for any $(x, y) \in R$, we have $|y| \le p(|x|)$.
  - At least writing down answers should be doable efficiently

The class PF

If you want to output y,
output $0y$
if you want to output $\perp$,
output 1

- $R \subseteq \{0,1\}^* \times \{0,1\}^*$; a search problem
- R is *efficiently solvable*, if there exists a *polytime algorithm* A such that for any $x \in \{0,1\}^*$, $A(x) \in R(x)$ if $R(x) \neq \emptyset$, and $A(x) = \perp$ if $R(x) = \emptyset$
- $\mathcal{PF} = \{R : R \text{ is polynomially bounded and efficiently solvable}\}$
  - **P**olynomial-time **F**ind

The class PF

P , NP

- Not a 'standard' class per se, but very reasonable
  - Typically, complexity theorists only deal with classes of decision problems
- Many examples
  - In fact, you have learned lots of examples already

The class PC

- $R \subseteq \{0,1\}^* \times \{0,1\}^*$: a polynomially bounded search problem
- R has *efficiently checkable solutions*, if there exists a polytime algorithm A, such that for any x, y, $A(x, y) = 1$ iff $(x, y) \in R$
- $\mathcal{PC} = \{R \,:\, R \text{ has efficiently checkable solutions}\}$
  - **P**olynomial-time **C**heck

The class PC

- Again, not a 'standard' class
- But, again a very reasonable class
- If $R \in \mathcal{PC}$, then
  - For some polynomial p s.t., for any $(x, y) \in R$, $|y| \leq p(|x|)$, and
  - Given x, y, it can be efficiently checked if $(x, y) \in R$ or not
- Many examples

The class PC

$x = \langle p \rangle$

$y = \langle \alpha \rangle$

$(x, y) \in R$ iff $p(\alpha) = 0$

- Examples
  - Solving a system of polynomial equations
  - Integer factorization $x = \langle n \rangle$, $y = \langle a \rangle$ $a \mid n$, $1 < a < n$.
  - Finding a Hamiltonian path (or cycle)
  - The Traveling Salesman Problem (TSP)
  - …

PF versus PC

- Do we have $\mathcal{PF} \subseteq \mathcal{PC}$?
  - Not necessarily, perhaps
  - But philosophically…
- Do we have $\mathcal{PC} \subseteq \mathcal{PF}$?
  - We don't know!
  - But, it seems *very* reasonable that $\mathcal{PC} \nsubseteq \mathcal{PF}$!

The class P

- $S \subseteq \{0,1\}^*$; a decision problem
- S is *efficiently solvable*, if $\exists$ a polytime alg. A s.t., for every x, $A(x) = 1$ iff $x \in S$
- $\mathcal{P} = \{S \,:\, S \text{ is an efficiently solvable decision problem}\}$
  - **P**olynomial-time

NP proof system

- $S \subseteq \{0,1\}^*$; a decision problem
- S has an *efficiently verifiable proof system*, if $\exists$ a polynomial p and a polytime algorithm V such that
  - Completeness: $\forall x \in S, \exists y \in \{0,1\}^*, |y| \leq p(|x|) \wedge V(x,y) = 1$
  - Soundness: $\forall x \notin S, \forall y \in \{0,1\}^*$, we have $V(x,y) = 0$
- So, $x \in S$ iff there exists a *short* y with $V(x,y) = 1$

V: NP verifier.

The class NP

NP certificate
NP proof
NP witness

- In such a situation, S has an NP-proof system
  - And V its verification procedure
  - For $x \in S$, a short y with $V(x, y) = 1$ is an NP-witness of x
- $\mathcal{NP} = \{S : S \text{ has an efficiently verifiable proof system}\}$
  - **N**ondeterministic **P**olynomial-time
  - Due to some historical reasons

Completeness? Soundness?

- These terminologies are from mathematical proof systems
  - Completeness: any true statement is a theorem
  - Soundness: any theorem is true (if x is not true, then x is not a theorem)

completeness: $\forall x \in S, \exists y \in \{0,1\}^*, |y| \leq p(|x|) \wedge V(x,y)=1$

Soundness: $\forall x, \exists y \in \{0,1\}^*, V(x,y)=1 \longrightarrow x \in S$

$\forall x$, if $x \notin S \longrightarrow \neg (\exists y \in \{0,1\}^*, V(x,y)=1)$

$\therefore \forall x \notin S, \forall y \in \{0,1\}^*, V(x,y)=0.$

PC to NP

- $R \subseteq \{0,1\}^* \times \{0,1\}^*$; a search problem in $\mathcal{PC}$
- $S_R := \{x : R(x) \neq \emptyset\}$
- Then, $S_R \in \mathcal{NP}$ $\quad \exists p,\ (x,y) \in R \rightarrow |y| \leq p(|x|)$.

proof) Since $R \in PC$, $\exists V$: polytime TM s.t. $V(x,y)=1$ iff $(x,y) \in R$

Then, V is a NP proof system for S_R.

Completeness : $x \in S_R \rightarrow \exists y,\ (x,y) \in R \rightarrow V(x,y)=1 \wedge |y| \leq p(|x|)$.

Soundness : $x \notin S_R \rightarrow \forall y,\ (x,y) \notin R \rightarrow V(x,y)=0$

$\Rightarrow S_R \in NP$.

NP to PC

- $S \in \mathcal{NP}$ V, p.
- V: the verification procedure for S
- Then, there's a search problem $R \in \mathcal{PC}$ such that $S = S_R$

proof) Let $R = \{ (x,y) \mid |y| \le p(|x|) \wedge V(x,y) = 1 \}$

Then, $R \in PC$.

Moreover, $S_R = S$

if $x \in S_R$, then $\exists y$, $|y| \le p(|x|) \wedge V(x,y) = 1 \Rightarrow x \in S$

if $x \in S$, then, $\exists y$, $|y| \le p(|x|) \wedge V(x,y) = 1 \Rightarrow x \in S_R$

P versus NP

$\mathcal{P} \subseteq \mathcal{NP}$, because, if $S \in \mathcal{P}$ then we can define V as

$V(x,y)$: compute whether $x \in S$ or not,

if $x \in S$ output 1

if $x \notin S$ output 0

$\rightarrow S \in \mathcal{NP}$

- We have $\mathcal{P} \subseteq \mathcal{NP}$
- Do we have $\mathcal{NP} \subseteq \mathcal{P}$?
  - Or not: $\exists S \in \mathcal{NP}$ such that $S \notin \mathcal{P}$?
- It is widely believed that $\mathcal{P} \neq \mathcal{NP}$

V : an $\mathcal{NP}$ proof system for S

Completeness: $x \in S \longrightarrow V(x, \varepsilon) = 1$

Soundness: $x \notin S \longrightarrow \forall y,\ V(x,y) = 0$

If $S \in NP$, in fact there could be many possible proof systems $V_1, V_2, \cdots$ which make S an NP problem.

For example, if V is an NP-proof system for S,

consider V'.

$$V'(x, y) = 1 \quad \text{iff} \quad y = y_1 y_2 \text{ with } |y_1| = |y_2| \text{ and } V(x, y_1) = 1.$$

$\to V'$ is an efficient verifier for S

Completeness: if $x \in S$, $\exists y_1$; short, $V(x, y_1) = 1$, then $V'(x, y_1 y_1) = 1$

Soundness: if $x \notin S$ $\forall y_1$, $V(x, y_1) = 0$. but, then $V'(x, y) = 0$ for all y.