Complexity Theory

Jan Křetínský

Chair for Foundations of Software Reliability
and Theoretical Computer Science
Technical University of Munich

Summer 2016

Based on slides by Jörg Kreiker

Lecture 15

Public Coins and Graph (Non)Isomorphism

Goal and Plan

Goal

- understand public coins and their relation to private coins
- get a reason why graph isomorphism might not be **NP**-complete

Goal and Plan

Goal

- understand public coins and their relation to private coins
- get a reason why graph isomorphism might not be **NP**-complete

Plan

- show that graph non-isomorphism has a two round Arthur-Merlin proof; formally: GNI $\in$ **AM**[2]
- show that this implies GI is not **NP**-complete unless $\mathbf{\Sigma_2^p} = \mathbf{\Pi_2^p}$

Agenda

- **IP** and **AM** – recap
- graph non-isomorphism as a problem about set sizes
- tool: pairwise independent hash functions
- an **AM**[2] protocol for GNI
- improbability of **NP**-completeness of GI

IP

Definition (IP)

For an integer $k \geq 1$ that may depend on the input size, a language L is in **IP**$[k]$, if there is a probabilistic polynomial-time TM V that can have a k-round interaction with a function $P : \{0,1\}^* \to \{0,1\}^*$ such that

- Completeness
  $x \in L \implies \exists P. Pr[out_V\langle V, P\rangle(x) = 1] \geq 2/3$
- Soundness
  $x \notin L \implies \forall P. Pr[out_V\langle V, P\rangle(x) = 1] \leq 1/3$

We define **IP** $= \bigcup_{c \geq 1}$ **IP**$[n^c]$.

- V has access to a random variable $r \in_R \{0,1\}^m$
- e.g. $a_1 = f(x, r)$ and $a_3 = f(x, a_1, r)$
- g cannot see r

$\Rightarrow$ $out_V\langle V, P\rangle(x)$ is a random variable where all probabilities are

AM

Definition (AM)

- For every k the complexity class **AM**$[k]$ is defined as the subset of **IP**$[k]$ obtained when the verfier's messages are random bits only and also the only random bits used by V.
- **AM** $=$ **AM**$[2]$

Such an interactive proof is called an Arthur-Merlin proof or a public coin proof.

Agenda

- **IP** and **AM** – recap ✓
- graph non-isomorphism as a problem about set sizes
- tool: pairwise independent hash functions
- an **AM**[2] protocol for GNI
- improbability of **NP**-completeness of GI

Recasting GNI

- let G_1, G_2 be graphs with nodes $\{1, \ldots, n\}$ each
- we define a set S such that
  - if $G_1 \cong G_2$ then $|S| = n!$
  - if $G_1 \ncong G_2$ then $|S| = 2n!$

Recasting GNI

- let G_1, G_2 be graphs with nodes $\{1, \ldots, n\}$ each
- we define a set S such that
  - if $G_1 \cong G_2$ then $|S| = n!$
  - if $G_1 \ncong G_2$ then $|S| = 2n!$
- idea: S is the set of graphs that are isomorphic to G_1 OR to G_2
- if $G_1 \cong G_2$, this set is small, otherwise not

Recasting GNI

- let G_1, G_2 be graphs with nodes $\{1, \ldots, n\}$ each
- we define a set S such that
  - if $G_1 \cong G_2$ then $|S| = n!$
  - if $G_1 \ncong G_2$ then $|S| = 2n!$
- idea: S is the set of graphs that are isomorphic to G_1 OR to G_2
- if $G_1 \cong G_2$, this set is small, otherwise not
- problem: automorphisms
  - an automorphism of G_1 is a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\pi(G) = G$
  - all automorphisms of graph G written $aut(G)$

The infamous set S

$$S = \{(H, \pi) \mid H \cong G_1 \text{ or } H \cong G_2, \pi \in aut(H)\}$$

The infamous set S

$$S = \{(H, \pi) \mid H \cong G_1 \text{ or } H \cong G_2, \pi \in aut(H)\}$$

- to convince the verifier that $G_1 \ncong G_2$ the prover has to convince the verifier that $|S| = 2n!$ rather than $n!$
- that is the verifier should accept with high probability if $|S| \geq K$ for some K
- it should reject if $|S| \leq \frac{K}{2}$

Agenda

- **IP** and **AM** – recap ✓
- graph non-isomorphism as a problem about set sizes ✓
- tool: pairwise independent hash functions
- an **AM**[2] protocol for GNI
- improbability of **NP**-completeness of GI

Hash functions

- goal: store a set $S \subseteq \{0,1\}^n$ to efficiently answer membership $x \in S$
- S could change dynamically
- $|S|$ much smaller than 2^m, possibly around 2^k for $k \leq m$

Hash functions

- goal: store a set $S \subseteq \{0,1\}^n$ to efficiently answer membership $x \in S$
- S could change dynamically
- $|S|$ much smaller than 2^m, possibly around 2^k for $k \leq m$
- to create a hash table of size 2^k
  - select a hash function $h : \{0,1\}^m \to \{0,1\}^k$
  - store x at $h(x)$
- collision: $h(x) = h(y)$ for $x \neq y$
- choosing hash functions randomly from a collection, one can expect h to be almost bijective if $|S|$ is app. 2^k

Pairwise independent hash functions

Definition

Let $\mathcal{H}_{m,k}$ be a collection of functions from $\{0,1\}^m$ to $\{0,1\}^k$. We say that $\mathcal{H}_{m,k}$ is pairwise independent if

- for every $x \neq x' \in \{0,1\}^m$ and
- for every $y, y' \in \{0,1\}^k$ and

$Pr_{h \in_R \mathcal{H}_{m,k}}[h(x) = y \wedge h(x') = y'] = 2^{-2k}$

- when h is choosen randomly $(h(x), h(x'))$ is distributed uniformly over $\{0,1\}^k \times \{0,1\}^k$
- such collections exist
- here: we only assume the existence

Agenda

- **IP** and **AM** – recap ✓
- graph non-isomorphism as a problem about set sizes ✓
- tool: pairwise independent hash functions ✓
- an **AM**[2] protocol for GNI
- improbability of **NP**-completeness of GI

Goldwasser-Sipser Set Lower Bound Protocol

- $S \subseteq \{0,1\}^m$
- both parties know a K
- prover wants to convince verifier that $|S| \geq K$
- verifier rejects with high probability if $|S| \leq \frac{K}{2}$
- let k be an integer such that $2^{k-2} < K \leq 2^{k-1}$

Goldwasser-Sipser Set Lower Bound Protocol

The following protocol has two rounds and uses public coins!

V
- randomly choose $h : \{0,1\}^m \to \{0,1\}^k$ from a pairwise independent collection of hash functions $\mathcal{H}_{m,k}$
- randomly choose $y \in \{0,1\}^k$
- send h and y to prover

P
- find an $x \in S$ such that $h(x) = y$
- send x to V together with a certificate of membership of x in S

V if $h(x) = y$ and $x \in S$ accept; otherwise reject

Why the protocol works?

Intuition: If S is big enough (non-isomorphic case) then the prover has a good chance to find a pre-image.

Why the protocol works?

Intuition: If S is big enough (non-isomorphic case) then the prover has a good chance to find a pre-image.

Formally:

- show that there exists a $\hat{p}$ such that
  - if $|S| \geq K$ then $Pr[\exists x \in S. h(x) = y]$ is greater than $\frac{3}{4}\hat{p}$
  - if $|S| \leq \frac{K}{2}$ then $Pr[\exists x \in S. h(x) = y]$ is lower than $\frac{\hat{p}}{2}$
- this is a probability gap which can be amplified by repetition
- one can choose $\hat{p} = \frac{K}{2^k}$

Putting it together

AM[2] public coin protocol for GNI

- compute S (automorphisms) as above
- prover and verifier run set lower bound protocol several times
- verifier accepts by majority vote
- using Chernoff bounds, this gives the desired completeness and soundness probabilities
- observe: only a constant number of iterations necessary which can be executed in parallel

$\Rightarrow$ number of rounds stays at 2

Details: Arora-Barak, section 8.2

Agenda

- **IP** and **AM** – recap ✓
- graph non-isomorphism as a problem about set sizes ✓
- tool: pairwise independent hash functions ✓
- an **AM**[2] protocol for GNI ✓
- improbability of **NP**-completeness of GI

Graph Isomorphism

Theorem

If GI $= \{\langle G_1, G_2\rangle \mid G_1 \cong G_2\}$ *is* **NP**-*complete then* $\mathbf{\Sigma_2^p} = \mathbf{\Pi_2^p}$.

What have we learnt?

- graph isomorphism is not **NP**-complete unless the (polynomial) hierarchy collapses
- public coins are as expressive as private coins
  - proof of GNI $\in$ **AM**[2] generalizes to **IP**$[k] =$ **AM**$[k+2]$ (without proof)
  - one can also show **AM**$[k] =$ **AM**$[k+1]$ for $k \geq 2$ (collapse)
  - also not shown: perfect completeness for **AM**
- Goldwasser-Sipser set lower bound protocol (which is in **AM**[2])
- hash functions as a useful tool

Up next: **IP** $=$ **PSPACE**