5 May 2025

# Program Checking.

## Plan

* Program Checking
* Announcements
* Matrix Multiplication

## In 4820

Key Aspect of Algo Design

Proof of Correctness!

## Outside of 4820

May encounter programs w/ bugs :(

## In 4820

Key Aspect of Algo Design

Proof of Correctness!

## Outside of 4820

May encounter programs w/ bugs ☹

Given a (possibly-buggy) program P, can we check if P is correct on a given input?

# Program Checking.

## Given

* program $P$ supposed to solve problem $\Pi$
* input $x$ and evaluation $P(x)$

**Check:** Is $P$ correct on input $x$?

# Program Checking.

## Given

* program $P$ supposed to solve problem $\Pi$
* input $x$ and evaluation $P(x)$

## Check: Is $P$ correct on input $x$?

* If $P(x) = \Pi(x)$, accept
* If $P(x) \neq \Pi(x)$, reject

# Announcements

* FINAL EXAM. 13 May 2025, 7p.
  ↳ Watch Ed for any announcements

* OH. Check the online calendar.

* Fill out Course Evals!

# Program Checking.

## Given

* program $P$ supposed to solve problem $\Pi$
* input $x$ and evaluation $P(x)$

**Check:** Is $P$ correct on input $x$?

* If $P(x) = \Pi(x)$, accept
* If $P(x) \neq \Pi(x)$, reject

Checker should run faster than re-solving $\Pi$.

# Matrix Multiplication

$$A \cdot B = C$$

$$\begin{bmatrix} \text{---} & A_1 & \text{---} \\ \text{---} & A_2 & \text{---} \\ & \vdots & \\ \text{---} & A_n & \text{---} \end{bmatrix} \cdot \begin{bmatrix} | & | & & | \\ B_1 & B_2 & \cdots & B_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} A_1 \cdot B_1 & A_1 \cdot B_2 & \cdots & A_1 \cdot B_n \\ A_2 \cdot B_1 & A_2 \cdot B_2 & & \vdots \\ \vdots & & \ddots & \\ A_n \cdot B_1 & \cdots & & A_n \cdot B_n \end{bmatrix}$$

Naive Algo. $\quad O(n^3)$ operations

Fastest Algo. $\quad O(n^\omega)$ operations

where $\omega < 2.373$

# Checking Matrix Mult.

Given:

* Program P, supposed to compute Mat. Mult
* $n \times n$ matrices $A, B$
* $P(A, B) \longrightarrow C$

Goal. Check if $C = A \cdot B$ without running Mat. Mult.

# Matrix - Vector Multiplication

$$\begin{bmatrix} — & M_1 & — \\ — & M_2 & — \\ & \vdots & \\ — & M_n & — \end{bmatrix} \cdot \begin{bmatrix} | \\ v \\ | \end{bmatrix} = \begin{bmatrix} M_1 \cdot v \\ M_2 \cdot v \\ \\ M_n \cdot v \end{bmatrix}$$

Naive Algo: $O(n^2)$ operations

Faster than Mat. Mult.

Can we check Matrix Mult using calls to Matrix-Vector Mult?

Key Fact

if $C = A \cdot B$, then

$$C \circ r = (A \cdot B) \cdot r$$

for all vectors $\vec{r}$.

## Key Fact

if $C = A \cdot B$, then

$$C \cdot r = (A \cdot B) \cdot r$$

for all vectors $\vec{r}$.

Try random $\vec{r}$?

# Random "spot checks" (aka Frievalds' Algo)

<u>Given</u>: $A$, $B$, and $C$ supposedly equal to $A \cdot B$

<u>Repeat T times</u>

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$
$$y \leftarrow B \cdot r$$
$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

ACCEPT

# Random "spot checks" (aka Frievalds' Algo)

**Given:** $A$, $B$, and $C$ supposedly equal to $A \cdot B$

**Repeat T times**

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$

$$y \leftarrow B \cdot r$$

$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

ACCEPT

---

**Claim:** for every $r \in \{0,1\}^n$ $\quad z = (A \cdot B) \cdot r$

Repeat $T$ times.

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

  $x \leftarrow C \cdot r$

  $y \leftarrow B \cdot r$

  $z \leftarrow A \cdot y$
- if $x \neq z$, REJECT

ACCEPT

---

Claim. If $C = A \cdot B$, then the checker ACCEPTS with probability 1.

Repeat $T$ times.

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

  $x \leftarrow C \cdot r$

  $y \leftarrow B \cdot r$

  $z \leftarrow A \cdot y$
- if $x \neq z$, REJECT

ACCEPT

---

Suppose $C \neq A \cdot B$.

What is the probability Checker REJECTs?

**Claim.** If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[ C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

**Claim.** If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[ C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

**Corollary.** If $C \neq A \cdot B$, then

$$\Pr\left[ \text{Checker ACCEPTs} \right] \leq \frac{1}{2^T}.$$

**Claim.** If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[ C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

---

**Pf.** By the Principle of Deferred Decisions.

**Claim.** If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[ C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

---

**Pf.** By the Principle of Deferred Decisions.

If $C \neq A \cdot B$, then there exists $i, j$ s.t.

$$C_{ij} \neq (A \cdot B)_{ij}$$

$\Longrightarrow$ the $j$th entry of $r$ is crucial.

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $U = \sum_{k=1}^{n} C_{ik} \cdot r_k$ $\quad$ $V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $U = \sum_{k=1}^{n} C_{ik} \cdot r_k$ $\qquad V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$

Imagine we "defer" flipping the $j^{th}$ bit of $r$ until all others have been flipped.

$r_1, r_2, \ldots r_{j-1}, r_{j+1} \ldots r_n$ $\qquad r_j?$ (still random)

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $U = \sum_{k=1}^{n} C_{ik} \cdot r_k$ $\qquad V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$

Imagine we "defer" flipping the $j^{th}$ bit of $r$ until all others have been flipped.

$$r_1, r_2, \ldots r_{j-1}, r_{j+1} \ldots r_n$$

$r_j$ ? (still random)

$U_{\setminus j} = \sum_{k \neq j} C_{ik} \cdot r_k$ $\qquad V_{\setminus j} = \sum_{k \neq j} (A \cdot B)_{ik} \cdot r_k$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}} [U \neq V] \text{ ?}}$$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}}[U \neq V] \text{ ?}}$$

Case 1 $\quad U_{ij} = V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] = \Pr[r_j = 1] = 1/2$$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}}[U \neq V] \text{ ?}}$$

<u>Case ①</u> $U_{ij} = V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] = \Pr[r_j = 1] = 1/2$$

<u>Case ②</u> $U_{ij} \neq V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] \geq \Pr[r_j = 0] = 1/2$$

So.

* Checker always ACCEPTS when $C = A \cdot B$

* Checker REJECTS w.p. $1 - 1/2^T$ when $C \neq A \cdot B$

* Checker runs $3T$ Matrix-vector mults.

So.

* Checker always ACCEPTs when $C = A \cdot B$

* Checker REJECTs w.p. $1 - 1/2^T$ when $C \neq A \cdot B$

* Checker runs $3T$ Matrix-vector mults.

e.g. $T = 10 \log n$.

- Checker runs in $O(n^2 \log n)$ time

- Makes mistake with probability at most $1/n^{10}$.

# Where do you go from here?

* More Algo? CS 6820
* More Hardness of Computation?
  - ↳ Complexity Theory CS 4814
  - ↳ Computability Theory CS 4810
* Both?
  - ↳ Cryptography CS 4830
  - ↳ Quantum Computing CS 4813

# Where do you go from here?

* More Algo? CS 6820
* More Hardness of Computation?
  - ↳ Complexity Theory CS 4814
  - ↳ Computability Theory CS 4810
* Both?
  - ↳ Cryptography CS 4830
  - ↳ Quantum Computing CS 4813

Thank you for a great semester!