CS598 Algebraic and Geometric Complexity Theory

Lecture 7 (2023-02-14)

Logistics: pset 2 out tomorrow

last lecture: - complexity of $\mathbb{L}$ primitive ops
- matrix mult - naive
  - block
  - Strassen

today: bilinear complexity

thm (Strassen): $n \times n$ matrix mult in $O(n^{2.81\ldots})$ alg ckt size

idea: recursion

non-trivial algo for 2×2 matrix mult

- 7 mults
- 18 add

vs

- 8 mult
- 4 add

(naive)

$\Rightarrow$ saving 1 mult $>>$ cost of 14 add

Q: formalize?

def: alg ckt C $U = V \times W$ mult in C is a **non-scalar mult** if $\deg v, w \geq 1$ $\equiv$ $v, w \notin \mathbb{F}$, otherwise is **scalar mult**. C **multiplication complexity** s if it uses $\leq s$ non-scalar multiplications

Q: mult complexity $\rightsquigarrow$ ckt size?

def: a straight line program is a sequence of operations $v_{i,j} - v_s$ where $v_i = \begin{cases} x_j \\ \alpha \in \mathbb{F} \\ v_j + v_k & j,k<i \\ v_j \times v_k \end{cases}$

the length is s, output is v_s.

lem: f has ckt size s iff f has SLP length s

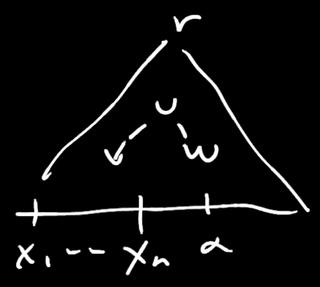
sketch: $\Leftarrow$: $v_1 \cdots v_j \cdots v_k \cdots v_i \cdots v_s$

$\Rightarrow$ all edges oriented in same direction

$\Rightarrow$ directed acyclic graph

$\Rightarrow$ ckt

$\Rightarrow$:



DAG

topological sort $\mapsto$ $v_1 \cdots v_n \; v_{n+1} \cdots v \cdots w \cdots u \cdots r$

$v_1 \cdots v_n \; v_{n+1}$

$\| \quad \| \quad \|$

$x \quad x_n \quad \alpha$

length

all edges oriented in same direction $\in$ SLP ∎

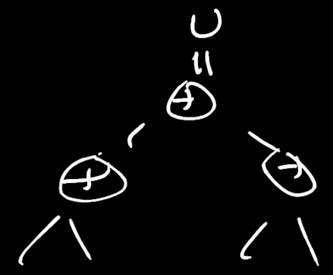
rmk: CKT $\equiv$ SLP, same results now appear in SLP

prop: f has circ C of mult complexity s

$\Rightarrow$ f has circ size $O((s+n)^2)$

of mult complexity $O(s)$

idea: C circ mult complexity s



v_1 -- v_m

$\leftarrow$ no non-scalar mult

but circ size m

$\Rightarrow$ no bound on size C in terms of s

eg: $x+1+1+\cdots+1 = x+m$

thm $\rightarrow$ is at worst a linear combination of other nodes $v_i = (\times)$

non-scalar mult

$O(s)$

pf: C straight line program for f

s non-scalar mult

claim: v_i **from** non-scalar mult in C

- v_i added
- v_i scalar mult

$\Rightarrow v_i = \sum_{l<i} \alpha_{il} \cdot v_l$

v_l non-scalar

$\cdot v_m$

$-1 \in F$

pf: by induction

$v_i = \beta \cdot v_j$: $j<i$

v_j = non-scalar mult : $v_i = \beta \cdot v_j$

$v_j = x_k$: $v_i = \beta \cdot x_k$

$v_j = \alpha$: $v_i = \alpha\beta \cdot 1$

else : $v_j = \sum_{l<j} \alpha_{jl} \cdot v_l$

$v_i = \beta \cdot \overrightarrow{v_l} = \sum_{l<j<i} \beta\alpha_{jl} \cdot v_l$

v_j

$v_i = v_j + v_k$

v_j — non-scalar
— var
— $\alpha \in \mathbb{F}$
— else

$\Rightarrow v_j \in \operatorname{span}_{v_\ell —}\{v_\ell\}_{\ell \le j < i}$

$\Rightarrow v_j \in \operatorname{span}_{v_\ell —}\{v_\ell\}_{\ell < j < i}$

rank

$v_j + v_k \in \operatorname{span}_{v_\ell —}\{v_\ell\}_{\ell < i}$ □

construct ckt C'

in C s non-scalr mult $v_i = v_j \times v_k$ $j,k<i$

$\Rightarrow \le O(s)$ other gates referenced

$\Rightarrow$ gate (rank) non-scalr mult v_ℓ

each a linear comb of { non-scalr gates v_p, $p<\ell$; vars ; 1 }

computed in $O(s+n)$ size
- $O(s+n)$ scalr mult
- $O(s+n)$ add

$\Rightarrow O(s(s+n)) = O((s+n)^2)$ additions

rank

rank: - required modification of ckt
- bounded ckt sz (within)

increase non-scalr multiplicns

Q: is this interesting?

A: $S \subseteq S^2$ wrt VP vs VNP

A: $S \not\subseteq S^2$ wrt matrix mult

n^3 vs $n^{2.81\ldots}$ vs $n^{2.373\ldots}$ vs ...

A: minimize nonscalar $\subseteq$ min ckt size

A: finding concrete complexity measure can be crucial

Q: multiplicative complexity of matrix mult?

def: a map $\mu : \mathbb{F}^n \times \mathbb{F}^m \to \mathbb{F}^p$ is **bilinear**

if it is linear in each input

- for fixed y, $x \mapsto \mu(x,y)$ is $\mathbb{F}$-linear

$\alpha, \beta \in \mathbb{F}$ $\quad \mu(\alpha x + \beta z, y) = \alpha \mu(x,y) + \beta \mu(z,y)$

- for fixed x, $y \mapsto \mu(x,y)$ is $\mathbb{F}$-linear

rmk: often $n = m = p$, but not always

eg: polynomial multiplication is bilinear

$\mu : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^{2n-1}$

$\Psi \qquad \Psi$

$f, g \mapsto f \cdot g$

$\deg < n$ (n coeffs) $\qquad \deg \le 2(n-1)$ ($2(n-1)+1$ coeffs)

$\alpha, \beta \in \mathbb{F}$, $(\alpha f + \beta h) g$

$= \alpha fg + \beta hg$ (distr law)

eg: matrix multiplication is bilin

$$\mu : \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times n} \to \mathbb{F}^{n\times n}$$

$$A, B \mapsto AB$$

$$(\alpha A + \gamma C)B = \alpha AB + \gamma CB$$

(distr)

bun: $\mu : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$ bilin

then $\mu_k(\bar{x}, \bar{y}) = \sum_{ij} \alpha_{ijk} x_i y_j$, $\alpha_{ijk} \in \mathbb{F}$

$\bar{x}, \bar{y} \in \mathbb{F}^n$ vectors in coord

bilin form

homog

pf: $\mu(\bar{x}, \bar{y}) \in \mathbb{F}^n$

$= \mu(\sum_i x_i \cdot \bar{e}_i, \sum_j y_j \bar{e}_j)$

standard basis vectors

$= \sum_i x_i \mu(\bar{e}_i, \sum_j y_j \bar{e}_j)$

$= \sum_i x_i \sum_j y_j \mu(\bar{e}_i, \bar{e}_j)$

$\mu(\bar{e}_i, \bar{e}_j) \in \mathbb{F}^n$

$= \sum_k \alpha_{ijk} \bar{e}_k$, $\alpha_{ijk} \in \mathbb{F}$ ∎

Q: what is complexity measure?

A: non-scalar mult

A: ——————, then

vespace then $\bar{x}, \bar{y}$

partition

eg: multiplication of linear polynomials

$aX+b,\ cX+d \mapsto acX^2+(bc+ad)X+bd$

$(a,b)\ (c,d) \mapsto$ ——————

computing $a\cdot c$ seems helpful

computing $a\cdot b$ seems unhelpful

def. $\bar{f} = f_1, \ldots f_n$ bilinear forms over $\mathbb{F}[\bar{x};\bar{y}]$ (n vars, n vars)

$f_k = \sum_{i,j} \alpha_{i,j,k}\, x_i y_j$

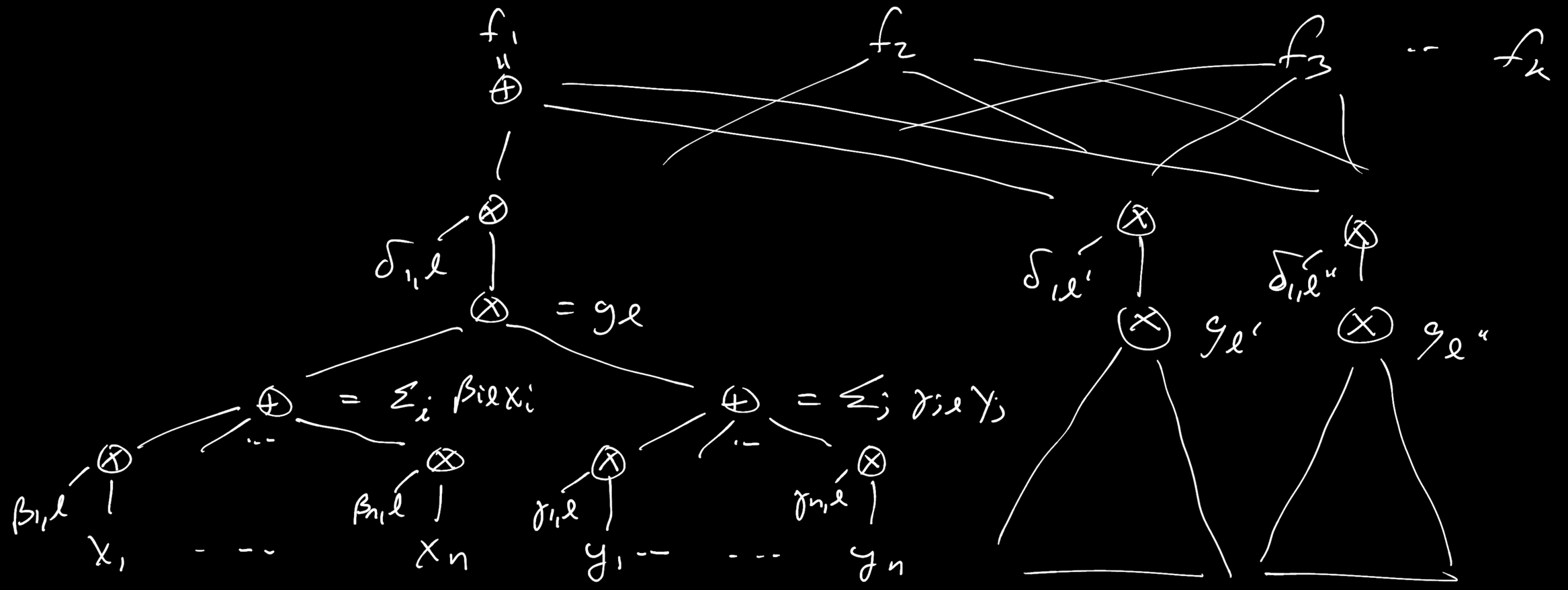
the **rank** of $\bar{f}$ is the min # of products

$g_\ell = \left(\sum_i \beta_{i,\ell}\, x_i\right)\left(\sum_j \gamma_{j,\ell}\, y_j\right)$

(non-scalar mults = expensive parts)

such that for each k, $f_k \in \mathrm{span}\{g_\ell\}_\ell$

$f_k = \sum_\ell \delta_{k,\ell}\, g_\ell$


f_1
f_2
f_3
f_4
$\delta_{1,\ell}$
$= g_\ell$
$= \Sigma_i \beta_{i\ell} x_i$
$= \Sigma_j \gamma_{j\ell} y_j$
$\beta_{1,\ell}$
$\beta_{n,\ell}$
$\gamma_{1,\ell}$
$\gamma_{m,\ell}$
x_1
x_n
y_1
y_n
$\delta_{1,\ell'}$
$\delta_{1,\ell''}$
$g_{\ell'}$
$g_{\ell''}$

Q: interesting?

prop. $\bar{f}$ bilinear forms over $F[\bar{x}, \bar{y}]$

(a) $\bar{f}$ has circuit of mult complexity $\leq \text{rank}(\bar{f})$

(b) ——— $s \Rightarrow \text{rank}(\bar{f}) \leq 2s$

correctness: clear

complexity: $O(1)$ total complexity per simulated gate

nonscalar mult only upper in

and they are bilinear

1 nonscalar mult $\mapsto$ 2 bilinear mult

pf: (a) clear

(b) by gate simulation. C circuit for $\bar{f}$

node v in C maps to $v_1, v_{\bar{x}}, v_{\bar{y}}, v_{\bar{x}\cdot\bar{y}}$ in C' $\Rightarrow$ discard higher degree terms

- v_1: constant term
- $v_{\bar{x}}$: linear in $\bar{x}$
- $v_{\bar{y}}$: linear in $\bar{y}$
- $v_{\bar{x}\cdot\bar{y}}$: bilinear in $\bar{x}, \bar{y}$

$v = v \times w$ nonscalar mult in C:

$v = v \times w$; scalar mult: clear

$v = v + w$: clear

$$v_{\bar{x}\cdot\bar{y}} = w_1 \cdot v_{\bar{x}\cdot\bar{y}} + v_1 \cdot w_{\bar{x}\cdot\bar{y}} + (v_{\bar{x}} \cdot w_{\bar{y}} + v_{\bar{y}} \cdot w_{\bar{x}})$$

homogenize

$$v_{\bar{x}} = v_1 \cdot w_{\bar{x}} + v_{\bar{x}} \cdot w_1$$

$$v_{\bar{y}} = v_1 \cdot w_{\bar{y}} + v_{\bar{y}} \cdot w_1$$

$$v_1 = v_1 \cdot w_1$$

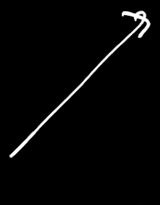
"thm": bilinear computation are optimal-ish

for bilinear problem

matrix mult

sketch: - non-scalar vs total

- bilinear vs non-scalar

today - 

- non-scalar mult vs total complexity
- bilinear rank vs multiplicative complexity

next lecture - tensor rank

logistics: pset 2 out tomorrow