A Supervisor's Reminiscence

What We Were Thinking

Albert R. Meyer

My concerns '67-'72:

(1) Was speedup ``real'' -- any sets with speedup not explicitly constructed (by diagonalization) to have it?

(2) Was there any ``real'' problem that was definitely not in P?

(3) How to defend a complexity theory which treated all finite problems as trivial?

Larrys' thesis provided definitive answers to all these questions.

A neat, but overlooked, result from Larry's thesis.

Language $\mathcal{L}$ has

IO-speedup

if, when *M* accepts $\mathcal{L}$, then

$\exists$ *M'* accepting $\mathcal{L}$, and

M' is very fast at ∞'ly many inputs where *M* is slow,

(same time on other inputs)

Pretty much all the problems known to be complete for the usual time and space bounded complexity classes have corresponding IO-speedup.

Cor: Let SF :=

{star-free reg. exps R

| $L(R) = \emptyset$}

SF has

$2^{2^{2^{\cdots n}}}$-IO-speedup

Definition: Let *M* be a program, *x* an input word, *f(n)* a time fcn. Say "*M(x)* is ***slow***" if on input *x*,

_M_ takes $>$ _f_(|_x_|) steps.

Definition: Let *M* be a program, *x* an input word, *f(n)* a time fcn. Say "*M*(*x*) is ***very fast***" if on input *x*,

M takes O(|x|) steps

(that is, linear time)

Definition: Let $\mathcal{D} \subseteq \Sigma^*$ be decidable.

$\mathcal{D}$ has ***f*-IO-speedup** if,

whenever L(*M*) = $\mathcal{D}$,

there is *M'* with L(*M*) = $\mathcal{D}$, and

for **infinitely many** *x*,

M'(x) is **very fast** while *M(x)* is **slow**

(*M'* takes same time as *M* at other *x*.)

Thm [Blum, McCreight-Meyer]

Let $\mathcal{D}_f :=$

$\{M \mid M$ rejects input "M" or takes $> f(|M|)$ steps$\}$

Then $\mathcal{D}_f$ has f-IO-speedup.

That is, $M \in \mathcal{D}_f$ iff

M rejects "M"

or is slow on it.

Pf: Suppose $\mathcal{D}_f = L(M_0)$. So

M_0 accepts "M " iff

M rejects "M " or is slow on it.

So, letting M be M_0,

M_0 accepts "M_0,"

but ***slowly***.

Cool Trick: let $M_{0,n}$ be same as M_0 but padded with n useless instructions.

So $M_{0,n}$ behaves exactly like M_0, just bigger.

So $M_{0,n}$ accepts "$M_{0,n}$" slowly,
so M_0 accepts "$M_{0,n}$" slowly.

Define M_0' to accept "$M_{0,n}$"
very fast for all n,
otherwise, same as M_0.

M_0' speeds up M_0.

QED

Lemma [Meyer/Stockmeyer]
IO-speedup inherits up
“efficient invertible“ reducibility.

r is "good" if it is 1-1, linear-bounded, "easy to compute," and the same for r^{-1}.

If $\mathcal{A}$ has *f*-IO-speedup,

and $x \in \mathcal{A}$ iff $r(x) \in \mathcal{B}$,

then $\mathcal{B}$ has $(\varepsilon \cdot f)$-IOspeedup

The story behind the SF result

My Background from Harvard ’62-‘67:

Recursion theory from Rogers & P. Fischer

$\leq_m$, arithmetic hierarchy

Complexity thy: Rabin, Hartmanis-Stearns, Blum

Speedup, compression, IO and AE

Finite Automata: S. Even, Ginsburg

Krohn-Rhodes on star-free events

Automata & Logic: Wang, Buchi

WS1S just another notation for FA’s

Efficient reducibility:

McCreight's thesis, CMU '68

Circuit complexity:

Winograd, Ehrenfeucht

Cook & Karp '72

Polytime $\leq$:

NP as $\exists$ polysize x

With this background, not surprising that I should propose a polytime analog of the arithmetic hierarchy.

But I wanted to show it was useful for classifying problems.

Suggested to Larry that he fit reg.exp. equivalence into the hierarchy.

Larry came back (next day?) and said reg.exp. equiv NOT IN the hierarchy! He showed me how to reduce every poly-class to reg.exp. equiv.

I pointed out that all he was using about poly-classes was they were in polyspace. He had shown reg.exp. equiv. was poly-space hard!

Driving home I started thinking ``why polynomial?''

Realized the crux was a linear size expression for set of all length *n* strings.

I knew how to get a nondeterministic FA to recognize all strings *differing* from the sequence of all consecutive *n*-bit strings. So with complement and a “smoothing” operation to turn one long string into all long strings, I could get a linear size expression for length 2^n strings, and could iterate the construction to get super-exponential lower bounds on reg-exps with complement & smoothing.

And the equiv. prob for these expressions was easy to reduce to the decidable logical theory WS1S, so it too had super-exponential lower bound.

WOW!

I called Mike Fischer (around midnight) to tell him what I'd figured out (and to make sure I wasn't kidding myself). Next day(?) I explained it to Larry and suggested he work on getting rid of the *ad hoc* ``smoothing'' operation. Later he got rid of ``star'' too.

We were off and running.