An introduction to computer science research: selected papers with commentary

Camille Akmut

Abstract
A survey of contemporary computer science literature and themes.

Introduction

Somewhere in the ‘Purgatory’ part of the Divine comedy, Dante wrote about that “the first age of mankind was gold”.

But, how will we one day look back on this “first age” of computer science and technology, and computer scientists and technologists?

Such is still uncertain, and undecided. We can only speculate.

Will the judgments of our successors be as harsh as those of Dante, who at his mildest wrote further that those “who wasted time ... must now pay with time”\textsuperscript{1}?

Dante however did not have only scorn and contempt for his contemporaries or predecessors, found in the various levels of hell and purgatory; in many other passages his efforts are entirely to the celebration of others, first among them his guide Virgil, whom he calls “gentle father”.

Through Statius, he continues to declare his love for Virgil and those like Virgil who “carried their lamps behind them, not for their own use or good but for others [to make others wise]”\textsuperscript{1}.

\textsuperscript{1}In the Purgatory still.
Programming languages


Description: a revolution in programming languages pedagogy but far from any nec plus ultra (“nothing further beyond”).


Description: introduction of a programming language “to be wondered at” (Latin, traditionally female given name).

— Wadler, Philip. 1987 [1986]. “A critique of Abelson and Sussman - or - Why calculating is better than scheming”

Description: the influential textbook Structure and Interpretation of Computer Programs would be better served by languages like “KRC or Miranda” (designed by David Turner and inspirations for Haskell) argues this author. Sussman, if not then already, later seemed to agree, who called Haskell “the most advanced” of the current, or “old languages”; though he did not give enough insight into how exactly the “new languages” should look like, or be.

\[
\text{sum} [\ {} ] = 0 \\
\text{sum} (x{:}xs) = x + \text{sum} \ xs
\]

Here is the same definition in Lisp:

\[
\text{(define (sum xs)} \\
\text{\quad (if (null? xs)} \\
\text{\quad \quad 0)} \\
\text{\quad (+ (car xs) (sum (cdr xs)))})
\]


Description: the so-called “sieve of Eratosthenes”, a method of determining prime numbers, can be found in various textbooks on (lazy) functional programming and yet, according to the author, this popular version is not the sieve. We propose that this variant can also be called instead “the common sieve (of Eratosthenes)”.

Algorithms


Description: In a modern language (more modern than ALGOL 60), in imperative style still:

```python
def shortestPath(m, n):

    INF = 10**10

    for i in range(1, n):
        for j in range(1, n):
            if m[j,i] < INF:
                for k in range(1, n):
                    if m[i,k] < INF:
                        s = m[j,i] + m[i,k]

    print(s)
```


Description: a contemporary survey from Dijkstra, Bellman and Floyd to present day improvements on them, and other solutions. Gives an insight into the kind of problems that computer scientists within this field of algorithms are concerned with (“putting another log in front of their algorithms” as we wrote...).

<table>
<thead>
<tr>
<th>Time Complexity</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n^3)</td>
<td>[52, 134]</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^{1/3})</td>
<td>55</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^{1/2})</td>
<td>126</td>
</tr>
<tr>
<td>(n^3/(\log n)^{1/3})</td>
<td>45</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^{2/3})</td>
<td>71</td>
</tr>
<tr>
<td>(n^3\log\log n/\log n)</td>
<td>127</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^{1/2})</td>
<td>141</td>
</tr>
<tr>
<td>(n^3/\log n)</td>
<td>27</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^{1/4})</td>
<td>72</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^2)</td>
<td>28</td>
</tr>
<tr>
<td>(n^3(\log\log n)/\log n^2)</td>
<td>73</td>
</tr>
</tbody>
</table>

Figure 1: The purgatory of “pure” computer scientists.
Cryptography / Security

Description: the original paper on power analysis (SPA and DPA) and breaking of cryptography; a form of so-called “side-channel attack” (in which hardware, not software properties or weaknesses are exploited). Contains, among many other things, a defense of multidisciplinary approaches in computer science (applicable to any science):

If cipher designers, software developers, and hardware engineers do not understand or review each other’s work, security assumptions made at each level of a system’s design may be incomplete or unrealistic.

— Halderman et al.. 2008. “Lest We Remember: Cold Boot Attacks on Encryption Keys”.
Description: the original paper on “cold boot attacks” i.e. computer memory (RAM) and breaking of cryptography (specifically disk encryption); another form of side-channel attack:

Contrary to popular assumption, DRAMs used in most modern computers retain their contents for several seconds after power is lost, even at room temperature and even if removed from a motherboard.

and the colder the temperature the longer the decay or remanence of data, or information.

— Muller, Tilo, Freiling, Felix and Dewald, Andreas. 2011. “TRESOR Runs Encryption Securely Outside RAM”.
Description: answer (and solution) to Halderman et al. 2008. If RAM is compromised, let us move encryption procedures to the CPU area instead.

Networking

Description: the original paper.

We present Tor, a circuit-based low-latency anonymous communication service. This second-generation Onion Routing system addresses limitations in the original design by adding perfect forward secrecy, congestion control, directory servers, integrity checking, configurable exit policies, and a practical design for location-hidden services via rendezvous points. Tor works on the real-world Internet (…)
Computer graphics

Description: “the legendary ‘inverse square root’”. See Sanjay Madhav’s Game Programming, pp. 46-47.

Artificial Intelligence

— Vinyals, Oriol and Le, Quoc. 2015. “A Neural Conversational Model”.
Description: an artificial intelligence is asked among other things “what is the purpose of life?” with surprising – sometimes contradictory, sometimes uneasy – answers and outcomes (“to live forever”).

Description: an introduction to this peculiar literature of most peculiar authors (and simultaneously contradiction). More than any of their colleagues perhaps these “most adroit” gentlemen and gentlewomen had “danced on all feet and to all songs”. Posterity is likely to be very harsh on them: they danced, we laughed.
(in order of appearance)

https://www.cs.kent.ac.uk/people/staff/dat/miranda/overview.pdf
https://www.cs.kent.ac.uk/people/staff/dat/miranda/wadler87.pdf
https://arxiv.org/abs/1705.02044
https://www.usenix.org/legacy/event/sec08/tech/full_papers/halderman/
halderman.pdf and https://citp.princeton.edu/topics/memory/
https://www.usenix.org/legacy/event/sec11/tech/full_papers/Muller.
pdf
https://www.freehaven.net/anonbib/cache/tor-design.pdf
https://dx.doi.org/10.17613/59fm-7y72
https://arxiv.org/abs/1506.05869