Introduction to computer science research : more papers discussed.

Camille Akmut

Abstract

Research from various fields within computer science is reviewed.

Algorithms

— Knuth, Donald. ongoing. *The Art of Computer Programming*.

Description : these volumes will be of different interests and uses to different people. Notable is the inclusion of passages called “History and bibliography” at the end of each section (before the exercises); In these, Knuth deploys enormous knowledge of his field to state exactly authors of algorithms, and their various circumstances. (As to the correctness of the provided information, others will decide but he must certainly be praised for at least attempting such a task : in his colleagues’ works, too often, algorithms seem to be the product of some immaculate conception, of no parents and no history...). Volume 1 (“Fundamental Algorithms”) and 3 (“Sorting and Searching”) appear as the most widely usable : they cover common data structures and algorithms (e.g. 2.2. lists, 2.3 trees, 5.2. sorting, etc.).

Computer graphics

— Newell, Martin. 1975. *The utilization of procedure models in digital image synthesis*. University of Utah. (PhD thesis.)

— ACM SIGGRAPH 1987. Conference proceedings in : *Computer Graphics* 21(4).

Description : the famous ‘Utah teapot’. Other such 3D test models or “benchmarks” include the ‘Cornell box’, ‘Stanford Bunny’, ‘Dragon’, OpenGL Gears... (Unfortunately neither this thesis nor this conference’s proceedings, where the teapot was widely featured, are readily accessible.)

(a) Original teapot at the Computer History Museum. (b) Toy Story (1995) at c. 15 min, by Pixar (co-founded by former Utah student).

Figure 1: The many lives of the ‘Utah teapot’.

Cryptography

— Perrin, Trevor and Marlinspike, Moxie. 2016. “The Double Ratchet Algorithm”.

Description : Specification. Sections 3 and 4 contain example Python implementations, 8 has useful references.

A KDF chain is a core concept in the Double Ratchet algorithm. (...) A KDF chain has the following properties (...) Forward security: Output keys from the past appear random to an adversary who learns the KDF key at some point in time. Break-in recovery: Future output keys appear random to an adversary who learns the KDF key at some point in time, provided (...)

— Cohn-Gordon et al. 2016. “On Post-compromise Security”.

In this work we study communication with a party whose secrets have already been compromised.

Description : By the same authors of “A Formal Security Analysis of the Signal Messaging Protocol”, where more information can be found.

Some more background : Gentlemen in this field of cryptography are both authors of some of the most exciting, practical and vitally important research in computer science, and creators simultaneously of some of the least consistent, coherent terminology encountered.

“Forward secrecy” means past messages cannot be decrypted in the event of a compromise (contrary to older systems like PGP, where a long-lived private key decrypts all messages). “Future secrecy” goes further by trying to achieve this but for future messages! (A ‘self-healing’ property.)

(The corresponding Computerphile video is entitled “Double Ratchet Messaging Encryption”. See 2:45-3:05 for a brief discussion of ‘forward security’, and 3:05 onwards ‘future security’.)

Programming languages

— **Rafkind, Jon and Flatt, Matthew. 2012.** “Honu: A Syntactically Extensible Language”.

Description : towards Scheme without parentheses and prefix (algebra) notation.

— **Doets and Eijck. 2004.** *The Haskell Road to Logic, Maths...*

— **O’Donnell et al.. 2006.** *Discrete Mathematics Using a Computer.*

Description : two innovative textbooks making use of a programming language (Haskell) to introduce mathematics.

CHAPTER 5. TREES

for instance is the (tree) representation of the formula :

$$(3 \times 4) + ((5 \times 6) \div 8)$$

```
Prelude> (3*4)+((5*6)/8)
15.75
```

— **Wirth, Niklaus. 2005.** “Good Ideas, Through the Looking Glass”.

[about OOP, Java and co.] *Nevertheless, the careful observer may wonder (...) Was this change of terminology expressing an essential paradigm shift, or was it a vehicle for gaining attention, a “sales trick”?*

Description : here Wirth looked back on half-a-decade of computer science culture. (I know that there are at least two versions of this : the one that was published, and another one containing additional passages – some of them extremely harsh and unforgiving.) Should prompt many to reconsider what, indeed, computer science is...

(in order of appearance, when available)

<https://signal.org/docs/specifications/doubleratchet/>

[/watch?v=9s02qdTci-s](#)

<http://www.cs.utah.edu/~rafkind/papers/honu-2012.pdf>

https://inf.ethz.ch/personal/wirth/Articles/GoodIdeas_origFig.pdf

Illustrations

Figure 1 (a) : Glydeck, <https://glydeck.blogspot.com/2011/08/who-remembers-utah-teapot.html>