Mihai Pătraşcu: Obituary and Open Problems

Mikkel Thorup

Figure 1: Mihai lived his short life in the fast lane. Here with his second wife Mirabela Bodic that he married at age 25.

Mihai Pătraşcu, aged 29, passed away on Tuesday June 5, 2012, after a 1.5 year battle with brain cancer. Mihai's academic career was short but explosive, full of rich and beautiful ideas as witnessed, e.g., in his 20 STOC/FOCS papers. His many interesting papers are available online at: `http://people.csail.mit.edu/mip/papers/index.html`.

Mihai's talent showed early. In high school he received numerous medals at national (Romanian) and international olympiads including prizes in informatics, physics and applied math. He received gold medals at the International Olympiad in Informatics (IOI) in both 2000 and 2001. He remained involved with olympiads and was elected member of the International Scientific Committee for the International Olympiad of Informatics since 2010.

Mihai's main research area was data structure lower bounds. In data structures we try to understand how we can efficiently represent, access, and update information. Mihai revolutionized and revitalized the lower bound side, in many cases matching known upper bounds. The lower bounds were proved in the powerful cell-probe model that only charges for memory access, hence which captures both RAM and external memory. Already in 2004 [17], as a second year undergraduate student, with his supervisor Erik Demaine as non-alphabetic second author, he broke the $\Omega(\log n/\log\log n)$ lower bound barrier that had impeded dynamic lower bounds since 1989 [6], and showed the first logarithmic lower bound by an elegant short proof, a true combinatorial gem. The important conclusion was that binary search trees are optimal algorithms for the textbook problem of maintaining prefix sums in a dynamic array. They also proved an $\Omega(\log n)$ lower bound for dynamic trees, matching Sleator and Tarjan's upper bound from 1983 [20]. In 2005 he received from the Computing Research Association (CRA) the Outstanding Undergraduate Award for best undergraduate research in the US and Canada.

I was myself lucky enough to meet Mihai in 2004, starting one of most intense collaborations I have experienced in my career. It took us almost two years to find the first separation between near-linear and polynomial space in data structures [19]. What kept us going on this hard problem was that we always had lots of fun on the side: playing squash, going on long hikes, and having beers celebrating every potentially useful idea we found on the way. A strong friendship was formed.

Mihai published more than 10 papers while pursuing his undergraduate studies at MIT from 2002 to 2006. Nevertheless he finished with a perfect 5.0/5.0 GPA. Over the next 2 years, he did his PhD at MIT. His thesis "Lower Bound Techniques for Data Structures" [11] is a must-read for researchers who want to get into data structure lower bounds.

During Mihai's PhD, I got to be his mentor at AT&T, and in 2009, after a year as Raviv Postdoctoral Fellow at IBM Almaden, he joined me at AT&T. We continued our work on lower bounds, but I also managed to get him interested in hashing which is of immense importance to real computing. We sought schemes that were both truly practical and theoretically powerful [15].

With his amazing energy and creative spirit, Mihai continued his work with many different collaborators on diverse topics. Data structure lower bounds, however, remained his core research area. In one of his favorite papers "Unifying the

landscape of cell-probe lower bounds" [14], he identified a whole new level of structure and connectivity in the field.

On January 1, 2011, Mihai was diagnosed with brain cancer. After a partially successful operation on March 21, 2011, anti-seizure medicine took away most of his energy and clarity, but he still had an amazing intuition. He wanted to work till the very end, even though this meant I had to push him to work in a wheelchair the last 4 months. Less than a month before his passing, he was notified that he was co-winner of the 2012 EATCS Presburger Young Scientist Award, recognizing his huge contribution to the field.

1 Open Problems

In the last weeks before Mihai passed away, we talked about what were the important challenges in data structures. Our experience with data structure lower bounds has been that the strongest lower bounds have not been for abstract problems, but rather for concrete well-known problems. The list below contains concrete problems that we believe capture types of lower bounds that have so far never been proved. They are ordered with the ones I believe to be the most significant first.

Problem 1: Deterministic dictionaries

One of the most fundamental data structures is a dictionary which allows us to store and look up information associated with keys. Dictionaries are often identified with hash tables, which are the most common way they are implemented, but they could be implemented with binary search trees, supporting each operation in $O(\log n)$. Dictionary operations are a bottleneck for many kinds of data analysis, including the processing of high volume data streams. They are also in the inner loops of many algorithms and have been central to computing as long as we have had computers.

For more than 50 years, we have had good randomized solutions. The space is linear in the number of stored keys, and we support both updates and look-ups in constant expected time [4]. Lots of solutions are known pushing the randomness around, e.g., in a breakthrough, Fredman, Komlos, and Szemerédi [5] proved that there are dictionaries with constant deterministic look-up time. Dietzfelbinger et al. [2] showed that this deterministic constant look-up time can be maintained with randomized updates in constant expected time. The randomization implies a fundamental unreliability. We have to be prepared for some updates being slow, which is a problem for time-critical systems. The probability of such events can be reduced at the expense of longer average update times. The big open question is

> Does there exist a perfect deterministic dictionary using linear space while supporting both lookups and updates in worst-case constant time?

Apart from RP$\stackrel{?}{=}$P, we thought of this as the most important derandomization problem left in theoretical computer science. The current positive deterministic results are quite far away. If we want constant query time, the best known update time is $O(n^\varepsilon)$ where ε is any positive constant [7]. If we are willing to settle for doubly-logarithmic query time, the update time can be improved to logarithmic [9]. If we want a joint bound for both lookups and updates, the best known bound is $O(\sqrt{\log n / \log\log n})$ [1]. The dream is to get down to constant time for both lookups and updates. Even with amortization, this would be a major breakthrough.

We believe that the true answer is negative, but how can we prove it? Often it is much easier to prove deterministic lower bounds than randomized lower bounds, but this does not give a separation unless the deterministic lower bounds are higher than the known randomized upper bounds. A separation for dictionaries would be extremely interesting, and likely have ramifications for many other problems.

Problem 2: Multiphase problem

In data structures, the largest proven lower bounds are polylogarithmic [8], and a major challenge is to prove polynomial lower bounds like $n^{\Omega(1)}$. Mihai proposed a very interesting line of attack via his so-called multiphase problem in [13].

Problem 3: Set Intersection

In [19], we proved a separation between near-linear and polynomial space, e.g., showing that certain queries that can be supported in constant time with $n^{1.001}$ space, require $\Omega(\log\log n)$ time with $n\log^{O(1)} n$ space. However, no such separation is known between space $n^{1.001}$ and space n^{100}. In [18], together with Roditty, we conjectured a concrete hard problem for space n^α for any $\alpha \in (1, 2]$. The hardness is all based on set intersection:

instance for preprocessing: The construction algorithm receives the n sets $S_1, ..., S_n \subseteq [u]$. In a *regular* instance, for some set size parameter $s \leq u$, each set has size at most s, and each element appears in ns/u sets.

query: for given $(i, j) \in [n]^2$, the boolean query is whether S_i intersects S_j.

The two obvious solutions are to either store all the (positive) answers in the preprocessing phase, or to simply store the sets directly and intersect them during the query. A popular belief, consistent with all current upper bound ideas, is

that in general there is no smooth trade-off between these two extreme types of solutions. In fact, the problem seems hard even for random instances where each S_i is a random subset of $[u]$ each of polylogarithmic size. Then the expected number of intersections is $\tilde{O}(n^2/u)$ and they can be represented in a hash table of this size. The following conjecture states that this is the best possible.

Conjecture 1. *Let a and b be sufficiently large constants. Consider regular set intersection instances with n sets, universe size u, and set size* $s = \log^a n$. *If a data structure with constant query time uses only* $O(n^2/(u \log^b n))$ *space and makes no false negatives, then for some set intersection instance, the fraction of false positives is* $\Omega(1)$ *over all* $\binom{n}{2}$ *possible queries.*

In [18] we used this conjectured hardness to prove hardness of distance oracles for graphs. Proving the conjecture would likely lead to lower bounds for many other problems.

Problem 4: Dynamic 2D Range Counting

In the 2D range counting problem, we have points in 2D. We want to query the number of points in a rectangle specified by the 4 corner coordinates. In a decision version, we may just ask if the parity is odd. In [10] Mihai proved that for the static problem where n points have to be represented with near-linear space, the query time is $\tilde{\Omega}(\log n)$. He believed that in the dynamic case, the update time would be $\tilde{\Omega}(\log^2 n)$. This would be the first superlogarithmic lower bound for a decision problem.

We note that Larsen [8] provided the $\tilde{\Omega}(\log^2 n)$ lower bound for the case where each point has an $O(\log n)$-bit weight, and where the query is about the range sum (not just the parity). However, such large weights can code more information than a memory address, and this is exploited heavily by the technique from [8].

Problem 5: Succinct Dictionary

In succinct data structures, we try to represent data using space close to the entropy H, yet provide efficient access. In his FOCS'08 best student paper [12], Mihai showed that for many data structure problems, we can pick an arbitrary parameter t, and represent the data using space $H + O(H/\log^t H)$ space, answering queries in time $O(t)$. For most of the problems, no such result was known for any $t > 2$. For some of the problems, he later presented matching lower bounds together with Viola [16]. One problem for which he could not find a matching lower bound was the dictionary problem, where we are given n elements from $[u]$. The entropy is $H = \log_2 \binom{u}{n}$. The question is if we can do better than the $H + O(H/\log^t H)$ space

from [12]? Inspired by our paper [3], Mihai was hopeful that much better bounds would be possible. In [3], for the so-called trits problem, we improved the space from $H + O(H/\log^t H)$ to $H + O(1)$ while maintaining constant query time.

References

[1] A. Andersson and M. Thorup. Dynamic ordered sets with exponential search trees. *Journal of the ACM*, 54(3), 2007. See also FOCS'96, STOC'00.

[2] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. M. auf der Heide, H. Rohnert, and R. E. Tarjan. Dynamic perfect hashing: Upper and lower bounds. *SIAM Journal on Computing*, 23(4):738–761, 1994. See also FOCS'88.

[3] Y. Dodis, M. Pătraşcu, and M. Thorup. Changing base without losing space. In *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, pages 593–602, 2010.

[4] A. I. Dumey. Indexing for rapid random access memory systems. *Computers and Automation*, 5(12):6–9, 1956.

[5] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with 0(1) worst case access time. *Journal of the ACM*, 31(3):538–544, 1984. See also FOCS'82.

[6] M. L. Fredman and M. E. Saks. The cell probe complexity of dynamic data structures. In *Proc. 21st ACM Symposium on Theory of Computing (STOC)*, pages 345–354, 1989.

[7] T. Hagerup, P. B. Miltersen, and R. Pagh. Deterministic dictionaries. *Journal of Algorithms*, 41(1):69–85, 2001.

[8] K. G. Larsen. The cell probe complexity of dynamic range counting. In *Proc. 44th ACM Symposium on Theory of Computing (STOC)*, pages 85–94, 2012.

[9] R. Pagh. A trade-off for worst-case efficient dictionaries. *Nordic Journal of Computing*, 7:151–163, 2000. See also SWAT'00.

[10] M. Pătraşcu. Lower bounds for 2-dimensional range counting. In *Proc. 39th ACM Symposium on Theory of Computing (STOC)*, pages 40–46, 2007.

[11] M. Pătraşcu. *Lower Bound Techniques for Data Structures*. PhD thesis, MIT, 2008.

[12] M. Pătraşcu. Succincter. In *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 305–313, 2008.

[13] M. Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *Proc. 42nd ACM Symposium on Theory of Computing (STOC)*, pages 603–610, 2010.

[14] M. Pătraşcu. Unifying the landscape of cell-probe lower bounds. *SIAM Journal on Computing*, 40(3):827–847, 2011. Announced at FOCS'08.

[15] M. Pătraşcu and M. Thorup. The power of simple tabulation-based hashing. *Journal of the ACM*, 59(3):Article 14, 2012. Announced at STOC'11.

[16] M. Pătraşcu and E. Viola. Cell-probe lower bounds for succinct partial sums. In *Proc. 21st ACM/SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

[17] M. Pătraşcu and E. D. Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*, 35(4):932–963, 2006. Announced at SODA'04 and STOC'04.

[18] M. Pătraşcu, L. Roditty, and M. Thorup. A new infinity of distance oracles for sparse graphs. In *Proceedings of the 53nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 738–747, 2012.

[19] M. Pătraşcu and M. Thorup. Time-space trade-offs for predecessor search. In *Proc. 38th ACM Symposium on Theory of Computing (STOC)*, pages 232–240, 2006.

[20] D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983. See also STOC'81.