# Formal Reasoning: The Elements of Scale

Peter O'Hearn
Facebook

Beginning with work on Separation Logic [7] and continuing with Facebook Infer [3], I've been working for many years with many colleagues on trying to scale formal reasoning. First, I concentrated on *scaling reasoning to large codebases*, then on *scaling to many people*, and finally I concentrated on trying to *scale the impact*. I've learnt lessons and formed new opinions as a result of this experience.

- It's possible to scale reasoning to large code in an automatic analysis, if the analysis works compositionally [2].
- It's important to deliver feedback to programmers in tune with their workflows. In one case, an automated analysis deployed in batch mode saw a 0% fix rate, where the same analysis deployed incrementally in code review (to catch regressions introduced by code modifications) saw a 70% fix rate [3].
- Incremental deployment paired with full automation helps scale to many people or many teams [5].
- Key benefits of formal reasoning – e.g., the ability to summarize many states or paths at once – hold for under-approximate reasoning (proving the presence of bugs) as well as over-approximate (proving absence) [8, 4].
- Especially for full functional correctness, we are lacking verification techniques which bring incremental value in a way that lets us scale impact: put a little more (human) effort in, get a corresponding (measurable) impact as your reward. I refer to this aspiration as *pay as you go verification*, to contrast it to a *pay upfront* approach where much investment is made specifying numerous interfaces and doing complete proofs before any payback results.

Perhaps the largest learning of all is the value in taking a people-oriented approach, where formal reasoning methods adapt to help programmers rather than (or in addition to) the other way around [1, 6].

## References

1. S. Blackshear, N. Gorogiannis, P. W. O'Hearn, and I. Sergey. Racerd: Compositional static race detection. *PACMPL*, 2(OOPSLA):144:1–144:28, 2018.
2. C. Calcagno, D. Distefano, P. W. O'Hearn, and H. Yang. Compositional shape analysis by means of bi-abduction. *J. ACM*, 58(6):26, 2011.
3. D. Distefano, M Fahndrich, F Logozzo, and P.W. O'Hearn. Scaling Static Analyses at Facebook. *Commun. ACM*, 62(8), 2019.
4. N. Gorogiannis, P. W. O'Hearn, and I. Sergey. A true positives theorem for a static race detector. *POPL*, 2019.
5. P. W. O'Hearn. Continuous reasoning: Scaling the impact of formal methods. In *LICS*, pages 13–25, 2018.
6. P. W. O'Hearn. Experience developing and deploying concurrency analysis at facebook. In *SAS*, 2018.
7. P. W. O'Hearn. Separation logic. *Commun. ACM*, 62(2):86–95, 2019.
8. P. W. O'Hearn. Incorrectness logic. *POPL*, 2020.