Related Work

Fernando Magno Quintão Pereira

What makes a great paper?

- Idea
- Evaluation
- Related Work

If you cannot convince referees that your paper is novel, then it is unlikely that the paper will be accepted.

Convince the Reader that the Problem is Important

negative issues. We introduce a precise and scalable *static* approach and tool, named SIERRA, for detecting Android event-based races. SIERRA is centered around a new concept

Such errors are pervasive and pernicious: a study of 18,000 fixed bugs in the Android platform and apps has revealed that 66% of the high-severity bugs are due to concurrency [33]. Android concurrency research has shown that the majority of Android race bugs are event-driven races [7, 19, 23]; per Maiya et al. [23], in Android apps, event-driven races are 4–7 times more frequent than data races.

And the Contribution Advances the Field

Hence there is a strong impetus for constructing tools that help find event-driven races in Android apps. To find such races, several *dynamic* detectors have been proposed, e.g., DroidRacer [23], CAFA [19], and EventRacer Android [7]. However, dynamic detectors have two main issues. First, due to their dynamic approach, they are prone to false negatives, i.e., miss actual bugs (in our experiments, EventRacer missed 25.5 out of 29.5 true races on average). Second, their effectiveness hinges on high-quality inputs that can ensure good coverage [6], as well as efficient ways to explore schedules.

To address these issues, we propose a static approach to event race detection. Android's concurrency model makes

Static Detection of Event-based Races in Android Apps, ASPLOS 2018

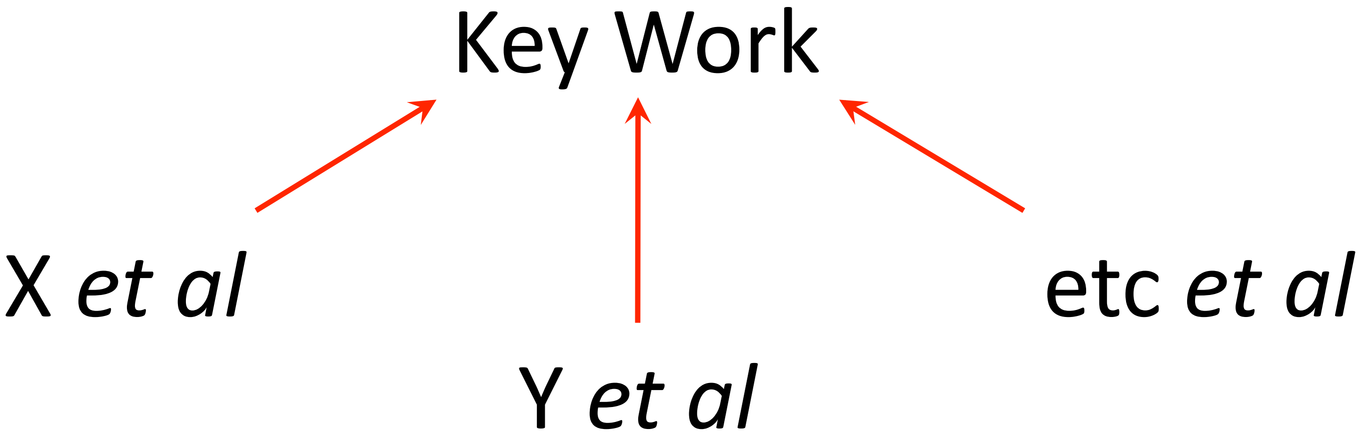
The Key Work

Closest Competitor

– or –

Origin of Everything

Backward Analysis

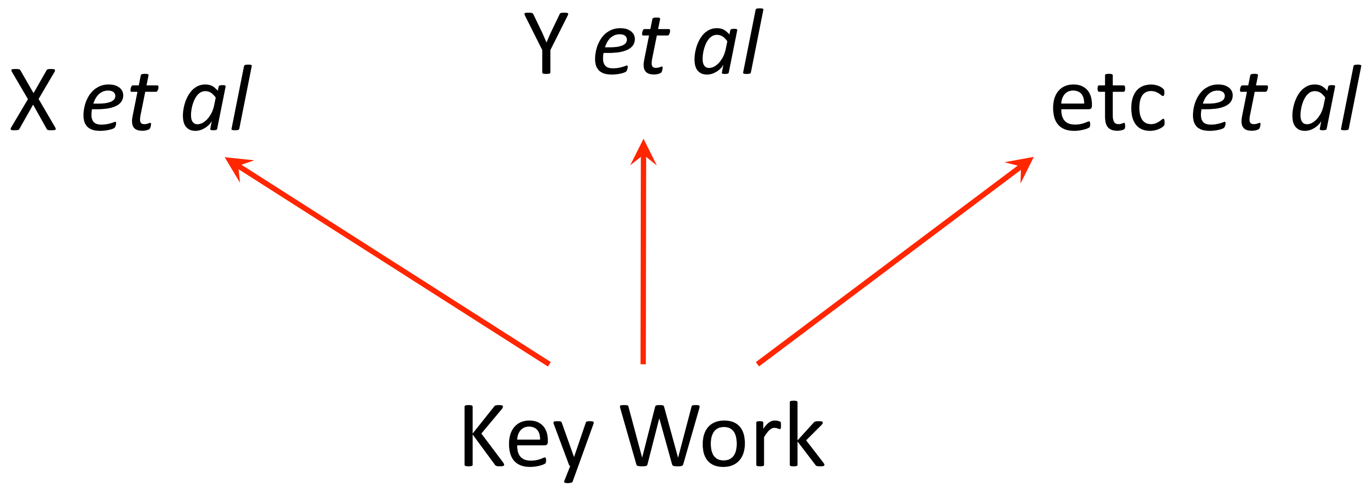


- Who started the field?
- What were the early challenges?
- What are the most influential works?

Example

The first work to deal with the pink analysis was due to U et al. After U et al' seminal paper, several research groups made sorties into this field. For instance, V et al have shown how to extend the early idea of U et al to handle also yellow cases. A few years later, W et al provided a way to make pink analysis faster. X et al also made advances in this direction, and for the first time made it possible to pinkfy very large programs. Y et al extended pink analysis to the world of distributed programs, and Z et al designed extensions to it to make it context sensitive. In those early days of pinkfication, most of the problems faced by researchers were related to scalability: the first implementations of pink analyses took a very long time to finish.

Forward Analysis



- Is the field still relevant?
- What are the current challenges?
- Who are my closest competitors?

The Core Question

How am I different from everything that already exists?

- Who are my closest competitors?

Example

In spite of intense research, pink analysis remains still a problem to be solved. Testimony of this fact is the number of related publications in the most relevant journals and conferences in the field of colors and hues. Today, most of the time and effort of researchers is directed towards precision: not every shade of pink can be satisfactorily identified. That is also the goal of this paper. Similar objectives have been pursed by A et al, B et al, C et al, and D et al. Our work is different from all of them in several ways. First, contrary to A et al and B et al, we are dealing with the pinkfication of Zebras, while they have designed techniques to pinkfy hummingbirds. In this sense, we face a scalability problem that is not present in the context of A et al and B et al. Moreover, contrary to C et al, we are using watercolor to perform the pinkfication. They use oil to thin their paint, and this decision leads to very different algorithms. Finally, this paper explores the use of two shades of pink in the same canvas - that is the main difference between our work and D et al's, who only uses one shade.

Check List

- Who started the field?
- What were the early challenges?
- What are the most influential papers?
- Is the field still relevant?
- What are the current challenges?
- Who are my closest competitors?
- How am I different from everything that already exists?

Keeping Track

- Get used to read papers regularly
- Summary the papers (RW.txt file, video, blog entry, etc)

```
* Automatic Pinkfication of Flying Hummingbirds, by A et al.
[Problem]
- Pinkfy a small object that moves fast and often duels among objects already pink.
[Contribution]
- On the fly pinkfication
- Use Markov Chains to predict next position of flying object
- Uses the fact that object moves in a relatively regular way
[Similarities]
- Same problem that we solve
- We also use Markov Chains
[Differences]
- Pinkfy a different object (Hummingbird vs Zebra)
- Does not use regular patterns in the target object
- Deal with much larger object
[Cite it]
"Contrary to A et al, we are dealing with the pinkfication of Zebras, while they have designed techniques
to pinkfy hummingbirds. In this sense, we face a scalability problem that is not present in the context of
A et al."
```