Resource-Parameterized Program Analysis using Observation Sequences

Peizun Liu

(under the supervision of Prof. Thomas Wahl)

Outline

- Overview of the Research
- A Paradigm
- Applications (What I Have Accomplished)
- How Might They be Relevant to Google

Outline

- Overview of the Research
- A Paradigm
- Applications
- How Might They be Relevant to Facebook

Problem Statement

Target is resource-parameterized programs, which are designed over a variable number of discrete resources.

“Resource” could mean:

Threads

Context Switches

Memory Writes

Execution

Message Queues

Problem Statement

Analysis is to ensure safety and liveness of such programs for an unspecified number of resource instances

Safety and liveness could mean:

- Free of data race / race condition in shared-memory multi-threaded programs
- Deadlock-free or mutual exclusion in concurrent programs
- Assertions in various sequential or concurrent programs
- Responsiveness in message-passing programs
- Eventual consistency in distributed systems
- ...

Why Do We Care?

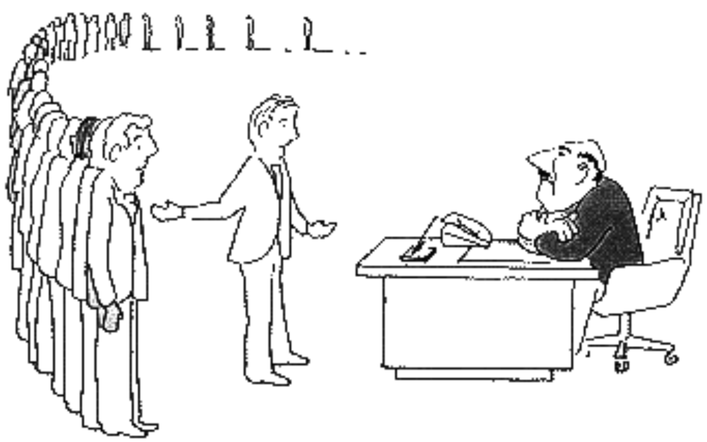
Reason 1: Resource-parameterized programs are ubiquitous.

Why Do We Care?

Reason 2: Ensuring their safety and liveness is desirable and significant.

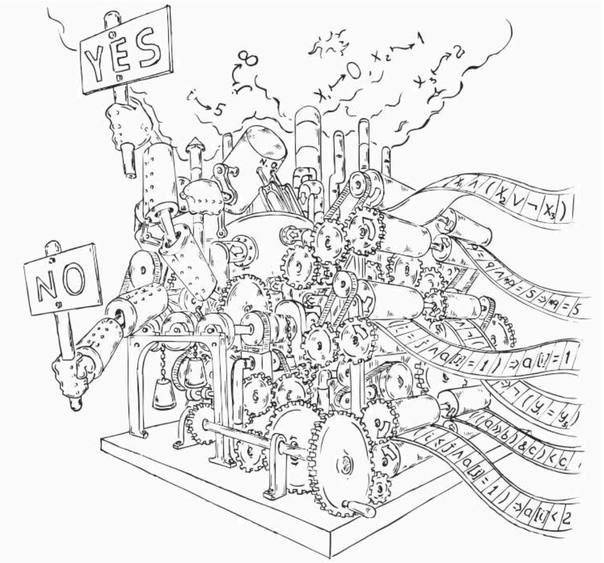
Why Do We Care?

However, resource-parameterized program analysis is challenging



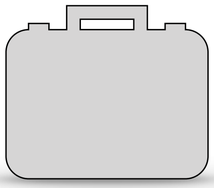
I can't find an efficient algorithm, but neither can all these famous people.

intractable

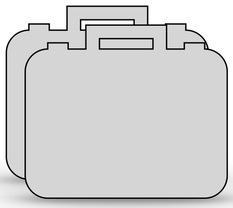


undecidable

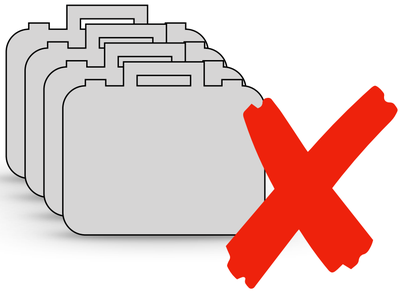
A Sidestep: Resource-Bounded Analysis



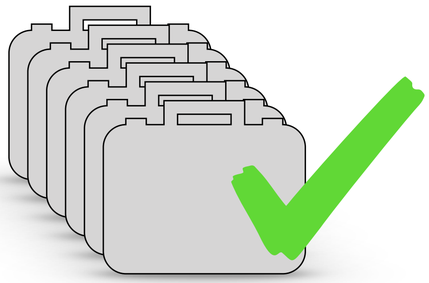
start with one resource



iteratively increase resource



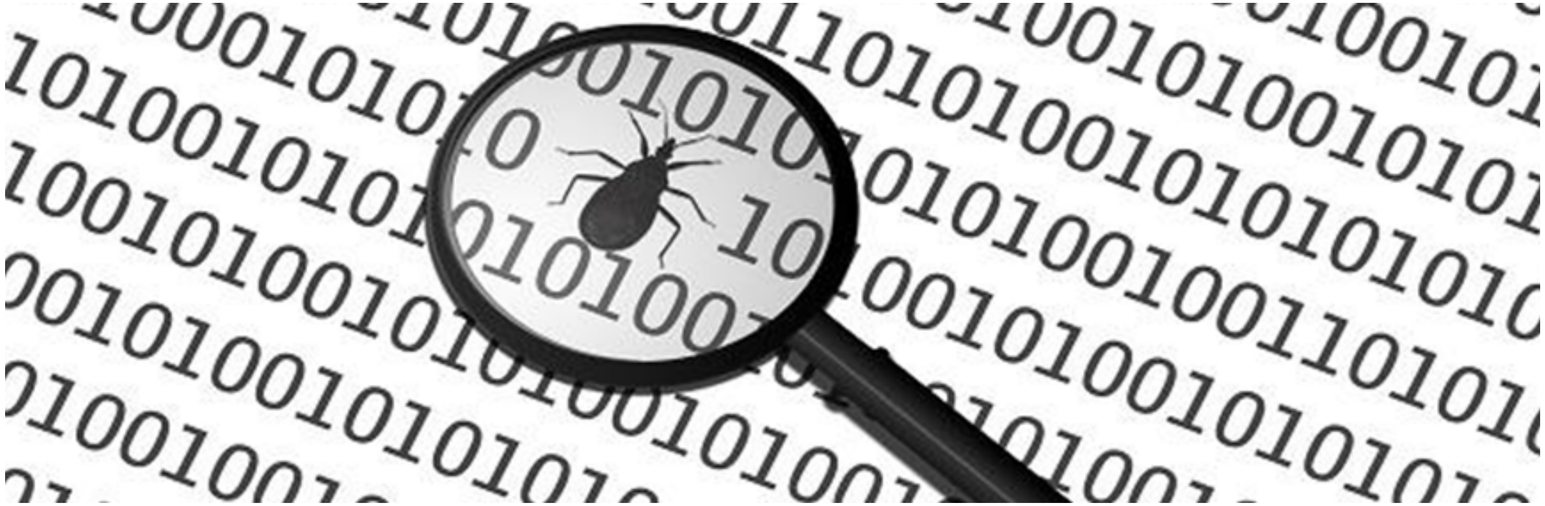
until a violation is found



until a pre-specified upper bound

A Sidestep: Resource-Bounded Analysis

We thus have a bug-finding technique.



A Sidestep: Resource-Bounded Analysis

Tested empirically [Lu, et. al. ASPLOS'08]: **Most bugs can be exposed with a small number of resources.**

Learning from Mistakes — A Comprehensive Study on Real World Concurrency Bug Characteristics

Shan Lu, Soyeon Park, Eunsoo Seo and Yuanyuan Zhou

Department of Computer Science,
University of Illinois at Urbana Champaign, Urbana, IL 61801
{shanlu,soyeon,eseo2,yyzhou}@uiuc.edu

A Sidestep: Resource-Bounded Analysis

Still, uncertainty remains beyond the pre-specified bound.

Beyond the Sidestep

Can we lift the resource-bounded bug-finding technique to resource-unbounded analysis?

Goal of the Research

To provide a uniform paradigm, which can

lift resource-bounded bug-finding technique to resource-unbounded analysis.

Our Paradigm: Bird's Eye View

Observation sequence (OS)

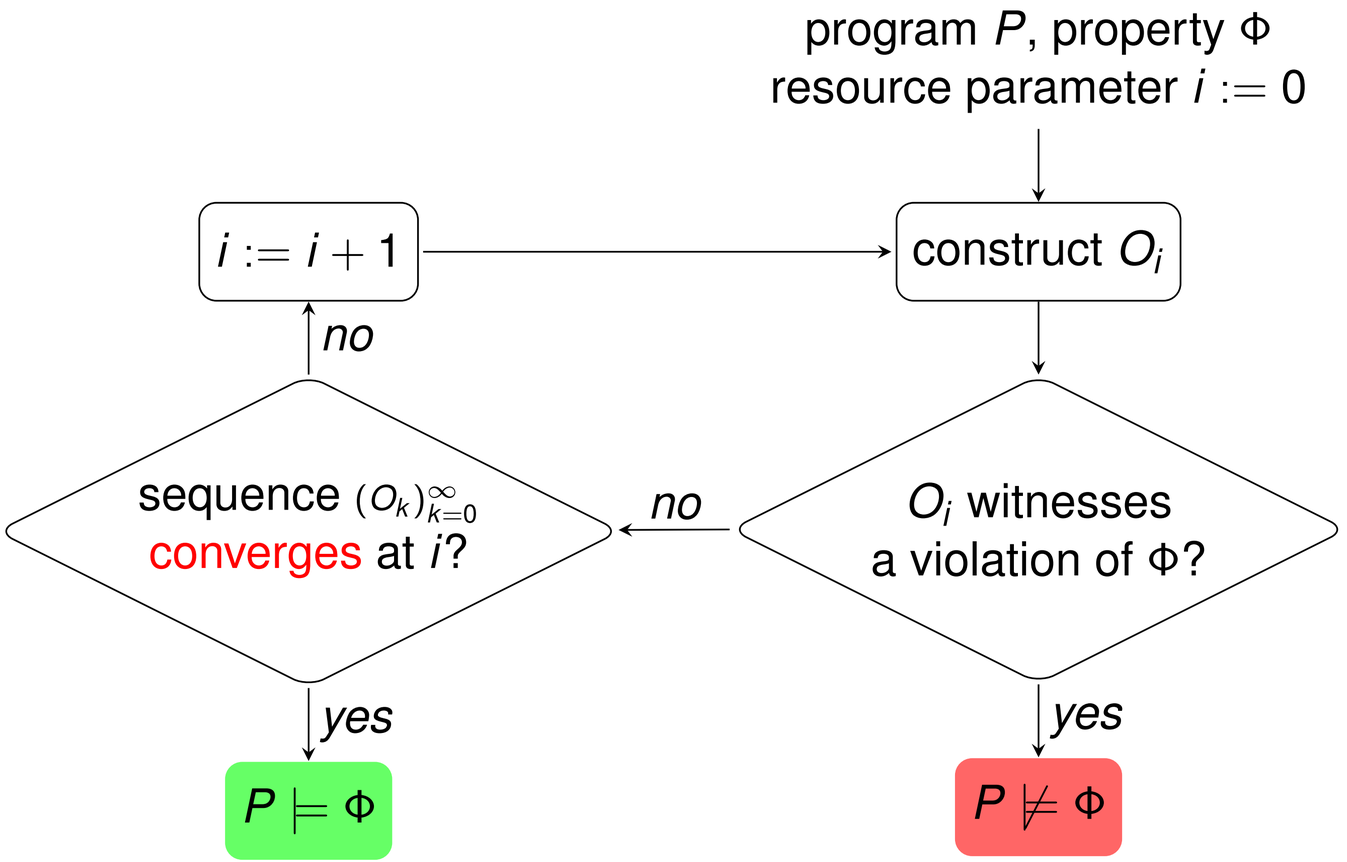
Informally, a sequence of observable program behaviors O_k within k instances of resource.

Examples:

- O_k := *{ reachable program states within k threads }*
- O_k := *{ reachable program locations within k context switches }*
- .....

Our Paradigm: Bird's Eye View

Scheme



Outline

- Overview of the Research
- A Paradigm
- Applications
- How Might They be Relevant to Facebook

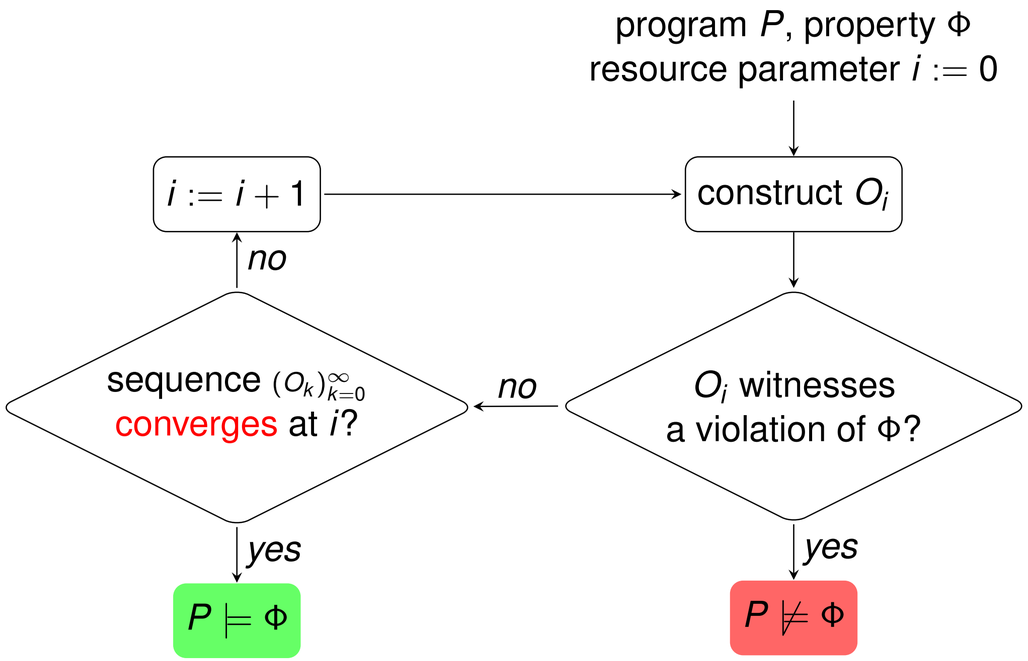
Observation Sequences

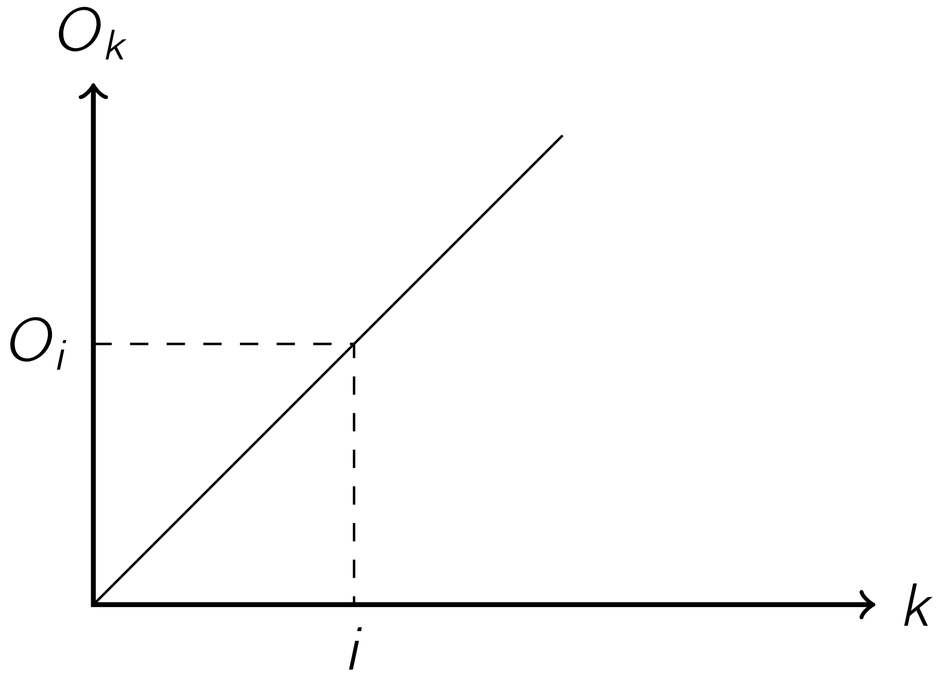
Definition

An observation sequence of a program P w.r.t. resource bound k is a sequence $(O_k)_{k=0}^{\infty}$ with the following properties:

- *for all k, $O_k \subseteq O_{k+1}$ that is monotonicity*
- *for all k, O_k is computable.*
- *for all k, $O_k \models \Phi$ is decidable, where Φ is a property of interest.*
- *for all k, $O_k \models \Phi$ implies $P \models \Phi$.*

Convergence Detection is Challenging

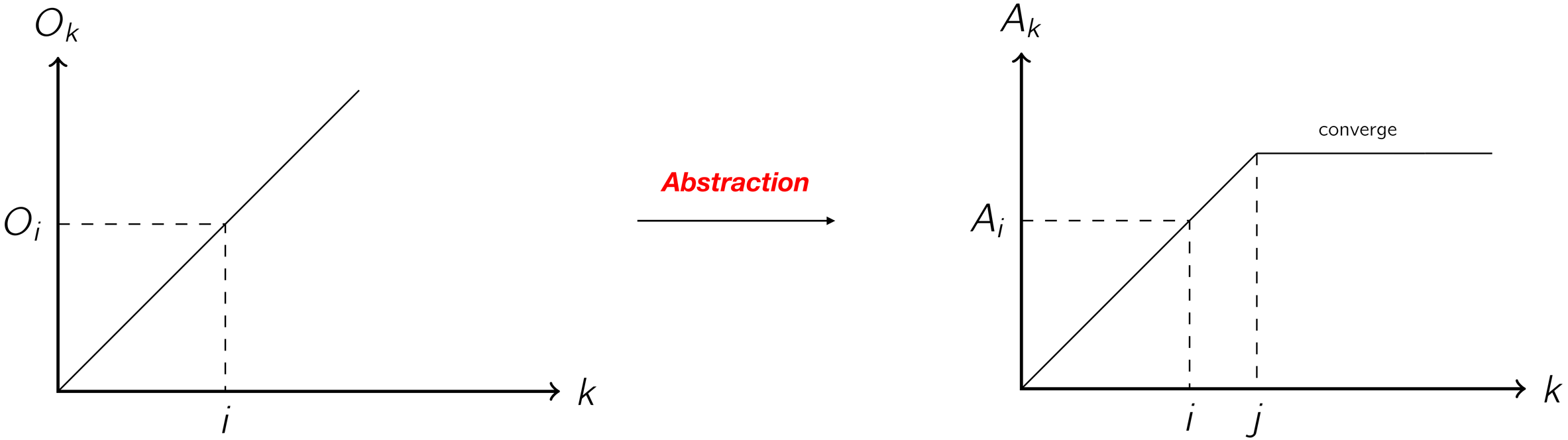




Convergence Property

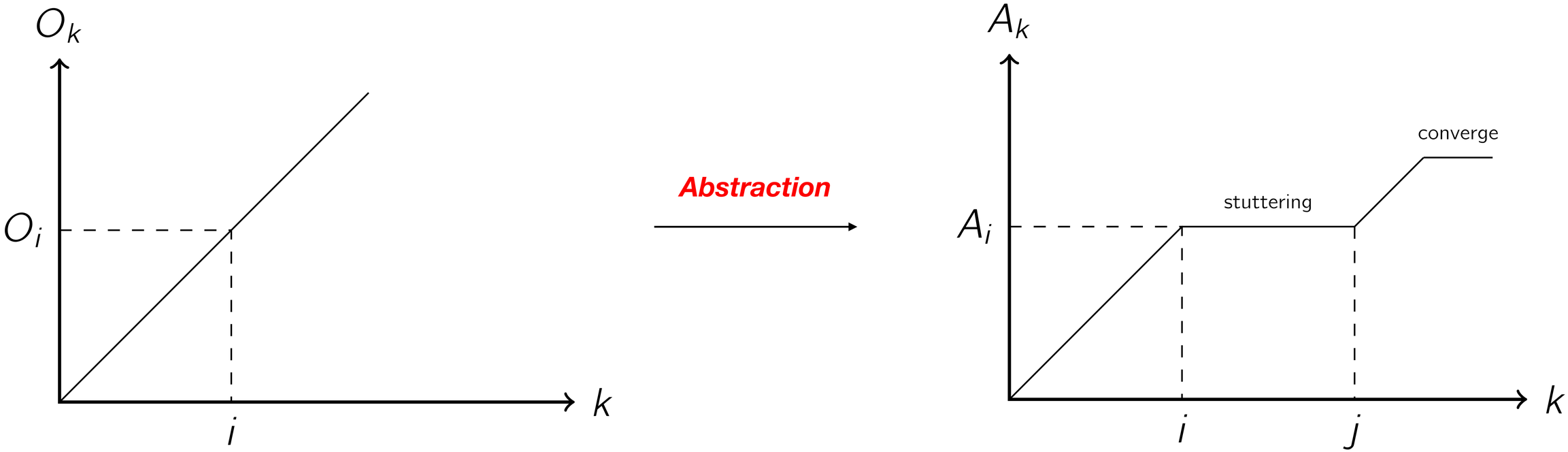
An OS $(O_k)_{k=0}^{\infty}$ over a finite domain always converges.

Abstraction



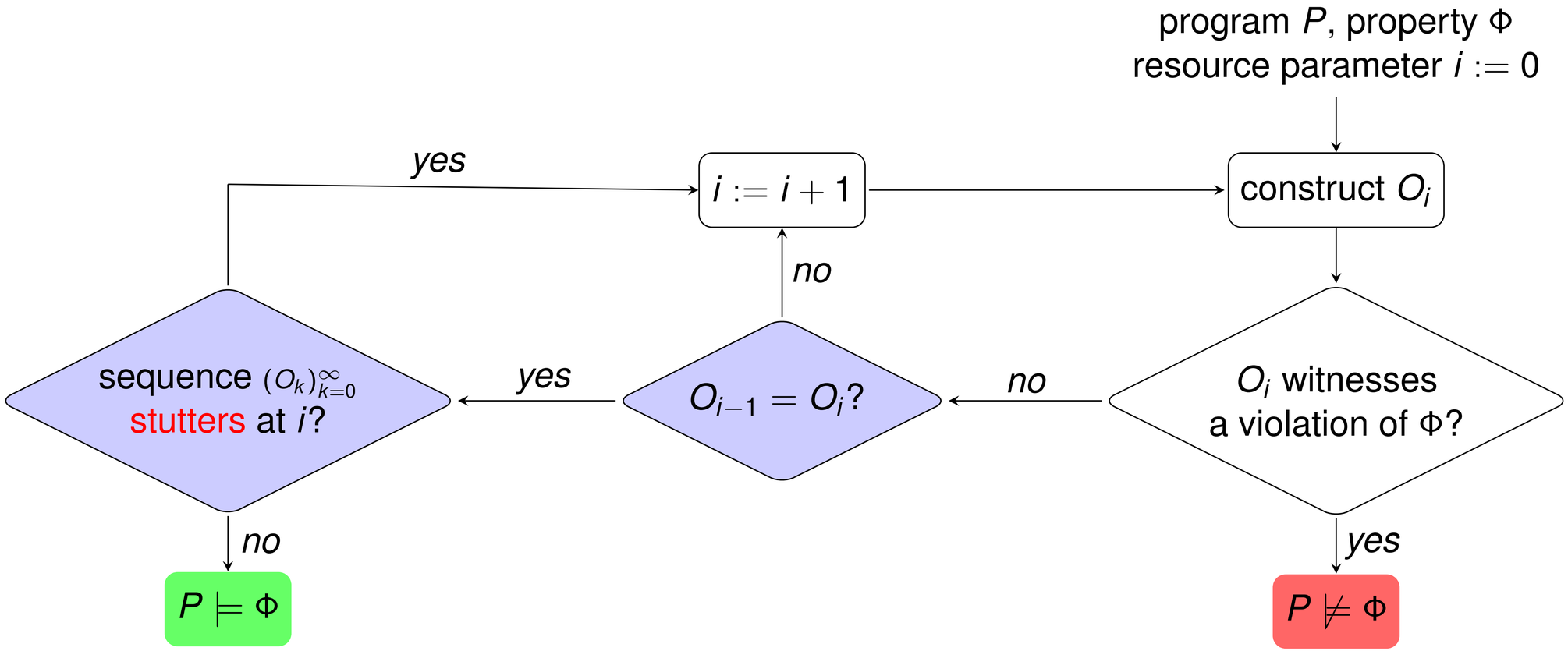
$$A_k := absrtaction(O_k)$$

Stuttering



$$A_k := absrtaction(O_k)$$

A Refined Scheme



Outline

- Overview of the Research
- A Paradigm
- Applications
- How Might They be Relevant to Facebook

Application 1: Thread-Parameterized Programs

Target is ...

shared-memory multi-threaded non-recursive programs.

Resource is ...

the number of threads in the executions.

Observation is ...

the set of reachable program states w.r.t. k threads.

Analysis is ...

to check the reachability of bad states.

Our Contributions to This Area

1. Peizun Liu and Thomas Wahl, "*Concolic Unbounded-Thread Reachability via Loop Summaries*". In ICFEM, pp.346-362, 2016.

2. Konstantinos Athanasiou, Peizun Liu and Thomas Wahl, "*Unbounded-Thread Program Verification using Thread-State Equations*". In IJCAR, pp. 516-531, 2016.

3. Peizun Liu and Thomas Wahl, "*Infinite-State Backward Exploration of Boolean Broadcast Programs*". In FMCAD, pp.155-162, 2014.

All tools and benchmarks are available on https://github.com/lpzun

Github

Application 2: Context-Parameterized Programs

Target is ...

shared-memory multi-threaded recursive programs.

Resource is ...

the number of context switches in the executions.

Observation is ...

the set of reachable program states w.r.t. k context switches.

Analysis is ...

to check the reachability of bad states.

Our Contributions to This Area

1. Peizun Liu and Thomas Wahl, "*CUBA: Interprocedural Context-Unbounded Analysis of Concurrent Programs*". In PLDI, pp.105-119, 2018.

2. Peizun Liu and Thomas Wahl, "*IJIT: An API for Boolean Program Analysis with Just-in-Time Translation*". In SEFM, pp.316-331, 2017.

Github

All tools and benchmarks are available on https://github.com/lpzun

Application 3: Queue-Parameterized Programs

Target is ...

message-passing programs and asynchronous event-driven programs.

Resource is ...

the size of message queues.

Observation is ...

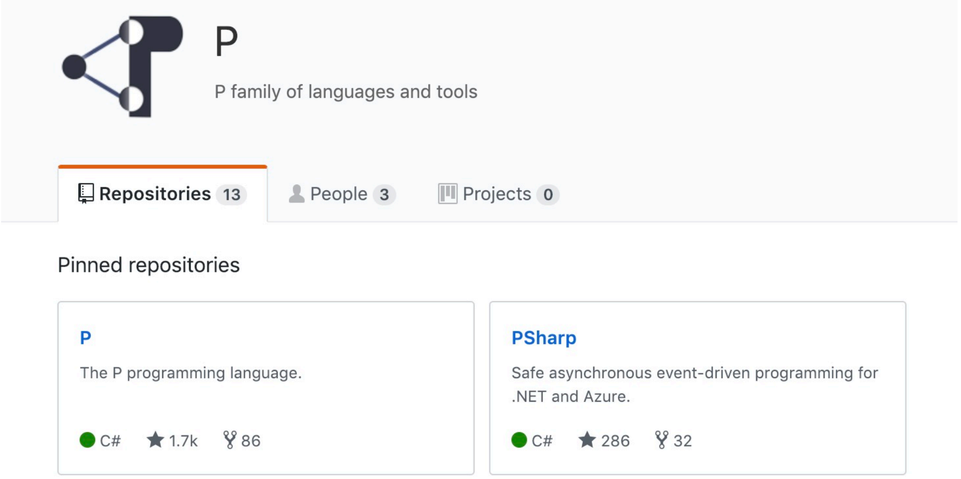
the set of reachable program states w.r.t. the size of queue within k.

Analysis is ...

to check the reachability of bad states.

Our Contributions to This Area

1. Peizun Liu, Akash Lal and Thomas Wahl, "*Verifying Asynchronous Event-Driven Programs Using Partial Abstract Transformers*". Under Submission.



Github

The tool and benchmarks is available on https://github.com/lpzun/P soon!

Take-Away

We target ...

resource-parameterized programs.

We proposed ...

a uniform paradigm of observation sequences.

The paradigm can ...

lift the bug-finding technique to resource-unbounded analysis.

We successfully applied ...

to verify several different resource-parameterized programs.

Limitations

- Scalability.
- Transition systems, how about programs?
- Benchmarks, application scenarios…

Outline

- Overview of the Research
- A Paradigm
- Applications
- How Might My Research be Relevant to Google

How Might My Research be Relevant to Google

- **Analysis of distributed systems and protocols**
  - Those include some crucial properties.
  - Model and verify them before ship them.
- **Analysis of various Android apps**
  - Help Android app developers to improve the reliability and security of their apps
- **Integrate the idea into existing static analysis tools**
  - Google-scale deep program analysis tools using abstract interpretation
  - …