# (Im)Possibilities of Predicate Detection in Crash-Affected Systems

Felix Gärtner

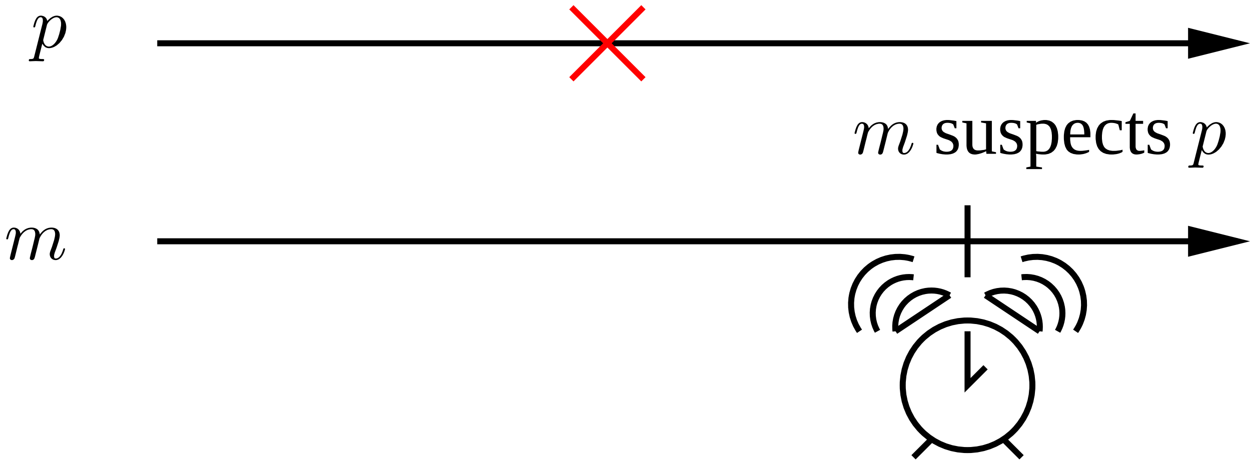
TU Darmstadt, Germany
felix@informatik.tu-darmstadt.de

joint work with Stefan Pleisch
IBM Research, Zurich Research Laboratory, Switzerland

# Recall WSS 1999 in Austin

- Session on future research directions.
- What has been the impact of stabilization research?
- How can we increase impact and awareness of stabilization research?
- Ted Herman's remark: Stabilization concepts are ubiquitous.
- Example: Unreliable failure detector abstraction by Chandra and Toueg [CT96].

# Failure Detectors



- Process $p$ is not suspected before it crashes.
- If $p$ crashes, it will eventually be suspected.
- Class of perfect failure detectors $\mathcal{P}$.

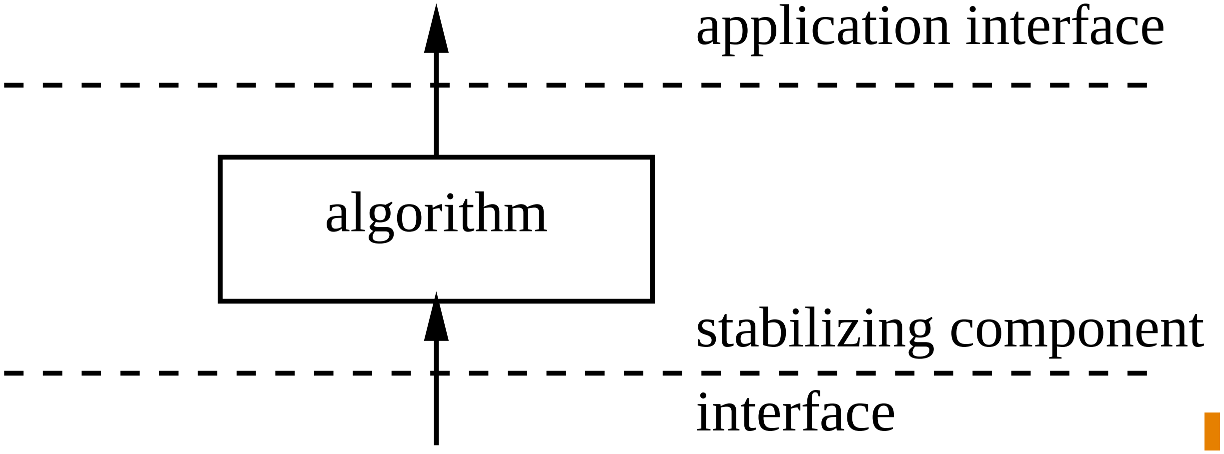
# Stabilizing Variants

- In practice “imperfect” failure detectors abound.
- Weaken the safety property
- Eventually perfect $\Diamond\mathcal{P}$ [CT96]:
  - Eventually no process is suspected before it crashes.
- Infinitely often accurate $\Box\Diamond\mathcal{P}$ [GM98]:
  - Correct application processes are not permanently suspected.

# Contributions to Stabilization Research

- Application: predicate detection (e.g. for debugging).

- Use stabilizing components to build reliable applications.

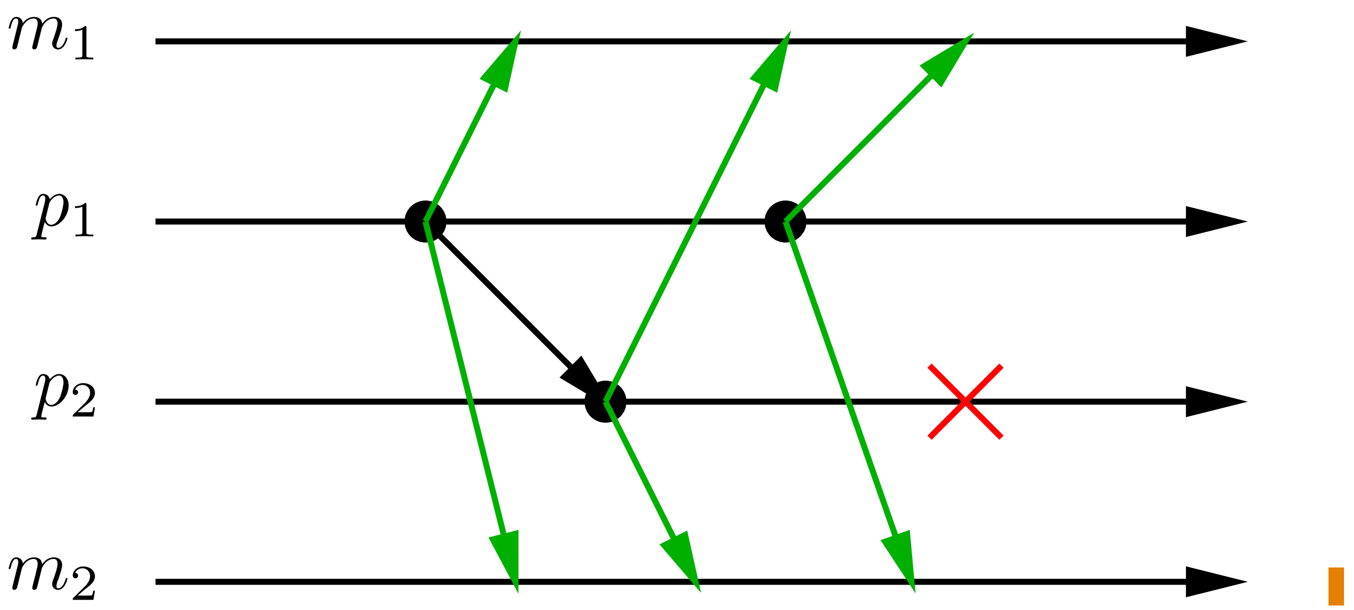


- How can we use (unreliable) failure detectors to build (reliable?) predicate detection algorithms?

# Outline

1. Three different predicate detection semantics.
2. Predicate detection algorithm.
3. Possibilities and impossibilities.
4. Advanced questions.

# Predicate Detection



- Does a global predicate $\varphi$ hold throughout the computation?

# Predicate Detection Semantics

- Perfect predicate detection $Sem_1$:
  - $(S)$ If the algorithm triggers a detection, then $\varphi$ has held in the computation.
  - $(L)$ If $\varphi$ holds, then the algorithm will eventually trigger a detection.

- Stabilizing predicate detection $Sem_2$: $L$ and $\Diamond S$.

- Infinitely often accurate predicate detection $Sem_3$:
  - $L$ and $\Box\Diamond S$.

# Application Example

- $\varphi \equiv$ "$p$ crashed while holding a lock"
- $Sem_1$ desirable (no wrong detections) but often infeasible.
- Finite number of wrong detections with $Sem_2$.
- If $\varphi$ never holds, $Sem_3$ excludes a "permanent" detection of $\varphi$.
- $Sem_3$ better than nothing.

# Algorithm

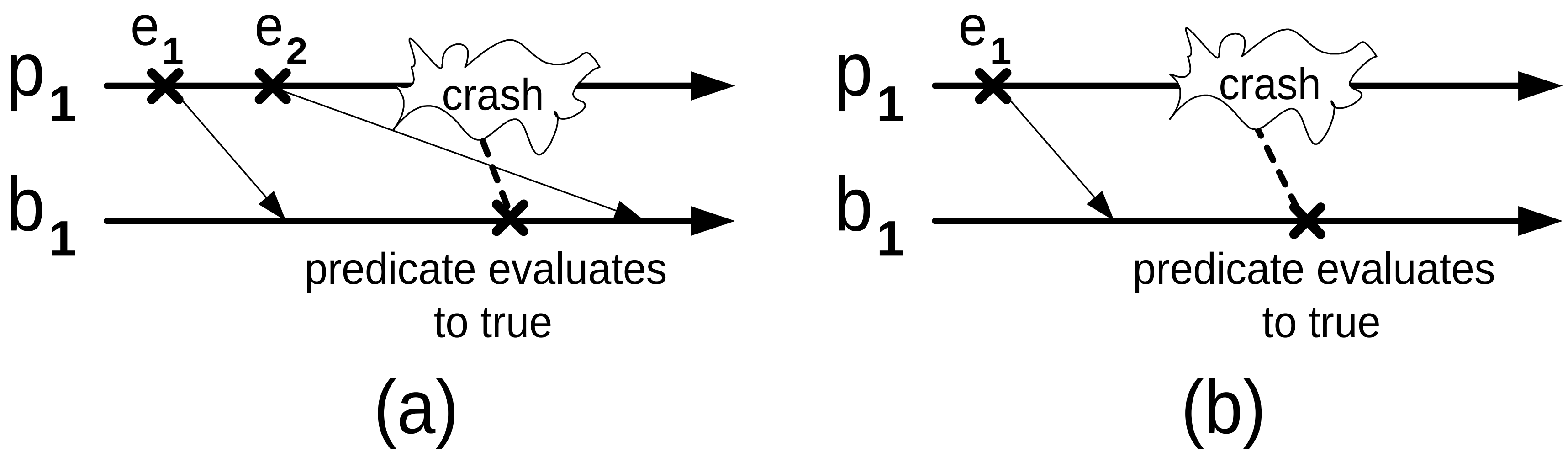
```
boolean variable $history$ initially false
upon $\langle$control message arrives or
        failure detector information changes$\rangle$ do
    $\langle$update own perception of global state$\rangle$
    if $\langle\varphi$ holds on global state$\rangle \wedge \neg history$ then
        $history := true$
        $\langle$trigger detection event$\rangle$
    elsif $\langle\neg\varphi$ holds on global state$\rangle \wedge history$ then
        $history := false$
        $\langle$trigger undetection event$\rangle$
    end
```

# Predicate Detection: Possibilities

- Stabilizing predicate detection $Sem_2$ achievable using $\Diamond\mathcal{P}$.
  - Eventual safety of $\Diamond\mathcal{P}$ leads to eventual safety of predicate detection.
- Infinitely often accurate predicate detection $Sem_3$ achievable using $\Box\Diamond\mathcal{P}$.
  - $\Box\Diamond\mathcal{P}$ avoids permanent “wrong detections”.

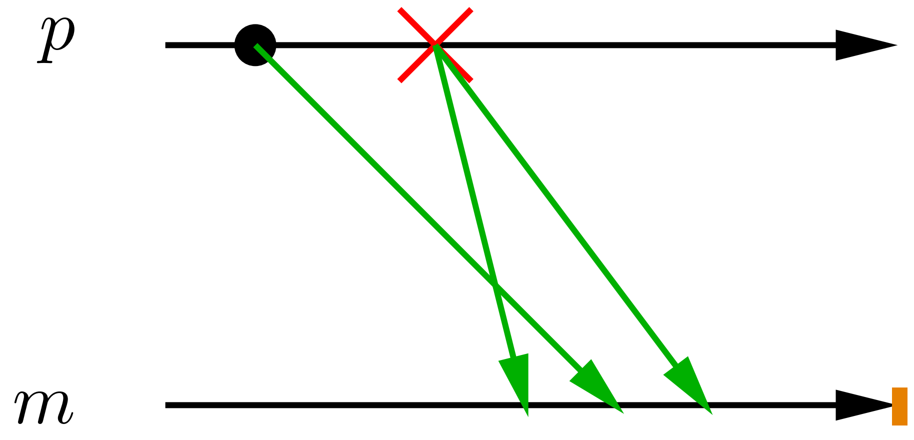
# Predicate Detection: Impossibilities

- $\mathcal{P}$ not sufficient for perfect predicate detection.



# Hypothetical Stronger Failure Detector

- Ordered perfect $\hat{\mathcal{P}}$:
  - Can relate crash event of process $p$ to the final event which happened on $p$.
  - Introduce a "visible" causality between control messages and failure detection events.



# Types of Failure Detectors

- Chandra and Toueg [CT96]: Query-style failure detectors.
  - Spurious detections can go unnoticed.
- Garg and Mitchell [GM98]: Interrupt-style failure detectors.
  - Every detection reaches application.
- We use interrupt-style ones: No difference for $\mathcal{P}$ and $\Diamond\mathcal{P}$.

# Stable and Observer Independent Predicates

- $\mathcal{P}$ is sufficient for $Sem_1$ if $\varphi$ is stable.
- If computation consists of more than one process:
  - Problems of observer dependence.
  - Introduce observation modalities [GK00] or restrict predicates to observer independent ones.
  - Only $\mathcal{P}$ and $\hat{\mathcal{P}}$ detect in an observer-independent way.

# Summary

- Predicate detection in crash-affected systems.
- Which predicate detection semantics are achievable using which types of failure detectors?
- Predicate detection is difficult even with perfect failure detectors.
- Must go for stabilizing predicate detection semantics in many practical settings.

# Acknowledgements

- Slides produced using pdfLATEX and Klaus Guntermann's PPower4.

## References

[CT96] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.

[GK00] Felix C. Gärtner and Sven Kloppenburg. Consistent detection of global predicates under a weak fault assumption. In *Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems (SRDS2000)*, pages 94–103, Nürnberg, Germany, October 2000. IEEE Computer Society Press.

[GM98] Vijay K. Garg and J. Roger Mitchell. Distributed predicate detection in a faulty environment. In *Proceedings of the 18th IEEE International Conference on Distributed Computing Systems (ICDCS98)*, 1998.