Motivation for Snapshot: Some Examples

- Write a distributed operating systems algorithm to detect whether an underlying user computation is deadlocked, or has terminated.

- The number of tokens in a computation never increases but may decrease. Design an algorithm to obtain an upper bound on the number of tokens.
Motivation Snapshots: Example

What happens when a computation dies while a snapshot is taken?

active

alive
died
terminated
dead

Time
Motivation Snapshots: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

- **alive**
  - Detection algorithm starts
  - Detection algorithm ends

- **dead**

Time
Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Should report: alive
Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that it is dead?

alive          dead

Detection algorithm starts          Detection algorithm ends

Time
Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Should report: dead
Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that it is dead?
Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Either
Specification of Detection Algorithms

Let P be a stable predicate of a system, i.e., stable(P)

Specification of an algorithm that detects P
1. The algorithm must terminate
2. If the algorithm terminates when NOT P holds then the algorithm must report that NOT P holds
3. If the algorithm starts when P holds then the algorithm must report that P holds.
4. (If the algorithm starts when NOT P holds and ends when P holds then it may report either.)
Motivation: Specification

reports dead implies dead now
alive now implies reported alive

Motivation: Specification

Time

Detection algorithm starts

Detection algorithm ends

alive now

reported alive
Specification of Detection Algorithm

- If the snapshot said that P holds then P holds now.

- If NOT P holds now then the snapshot said that NOT P holds.
More general problem

Given

\[ P \text{ next } (P \text{ OR } Q) \]

where \( Q \) is an operating systems action, and \( P \) is in the underlying computation,

Write an OS algorithm to detect \( P \)

Example:

\( P \): database computations are deadlocked
\( Q \): transactions have been aborted to break deadlock
Detecting a Stable Predicate $P$

```java
while( NOT P) {
    P = detectionAlgorithm();
    sleep(T);
}

// P holds
Q

// P may no longer hold
```
GENERAL APPROACH TO DETECTION ALGORITHMS
State of channel from R to P at time T (assume only one such channel)

State of channel from P to Q at time T (assume only one such channel)
Consistency of Local Snapshots

There exists a system computation in which all events before snapshot occur before all events after snapshot.
Pictorial Explanation: Consistency means the line can be straightened.
Pictorial Explanation

Events BEFORE snapshot

Events AFTER snapshot

No line crosses barrier from “after” to “before”
Key Idea of Proof

There exists a computation from the start state to the snapshot state, and from the snapshot state to the final state.
Key Idea of Proof

If a stable predicate holds in any state then it holds in all states reachable from that state.
If P holds in the start state (Stable P) then

If a stable predicate holds in any state then it holds in all states reachable from that state.
If P holds in the start state (Stable P) then

If P holds in the start state then it holds in the snapshot state
If $P$ holds in the start state (Stable $P$) then

If $P$ holds in the snapshot state then it holds in the finish state
If NOT P holds in the finish state then ...

If a stable predicate holds in any state then it holds in all states reachable from that state.
If NOT P holds in the finish state then it holds in the snapshot state which implies that it also holds for the start state.
Global Snapshot: Prevents this situation

Error: message sent after snapshot received before snapshot

Snapshot line

Message line
Marker message: Prevents this situation

Correct: message sent after snapshot received after snapshot
Determining messages in flight

What messages sent before snapshot are received after snapshot?

Message received after snapshot

Marker line

Message sent before snapshot
Determining messages in flight

marker line Q to P

Marker line P to Q

Message sent before snapshot

msgs in flight
When an agent takes a local snapshot it sends a marker on each of its outgoing channels.

When an agent receives a marker, the agent takes a local snapshot if it hasn’t done so already.

The messages in flight along a channel c to an agent Q are the messages received by Q after Q takes its snapshot and before Q receives a marker along c.