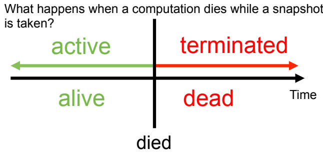
Motivation for Snapshot: Some Examples

- 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.

1

Motivation Snapshots: Example

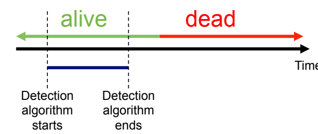
What happens when a computation dies while a snapshot is taken?



2

Motivation Snapshots: Specification?

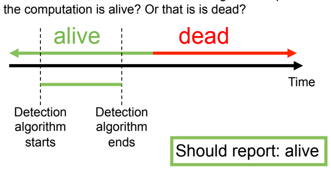
In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?



3

Motivation: Specification?

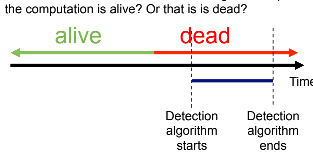
In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?



4

Motivation: Specification?

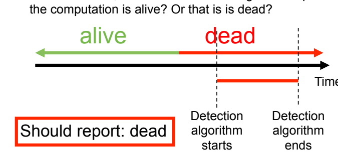
In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?



5

Motivation: Specification?

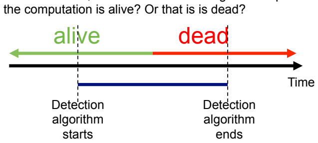
In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?



6

Motivation: Specification?

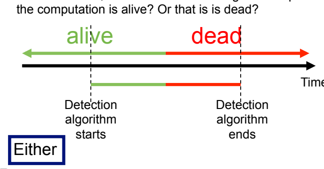
In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?



7

Motivation: Specification?

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



8

Specification of Detection Algorithms

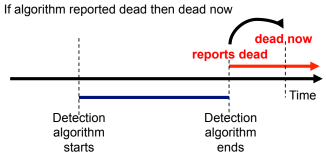
- Let P be a stable predicate of a system, i.e., stable(P)
- Specification of an algorithm that detects P

1. The detection algorithm must terminate
2. If the detection algorithm ***terminates*** when NOT P holds then the algorithm must report that NOT P holds
3. If the detection algorithm ***starts*** when P holds then the detection algorithm must report that P holds.
4. (If the detection algorithm starts when NOT P holds and ends when P holds then it may report either.)

9

Motivation: Specification

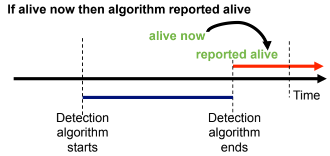
If algorithm reported dead then dead now



10

Motivation: Specification

If alive now then algorithm reported alive



11

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.

12

More general problem

Given

P next (P 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

13

Detecting a Stable Predicate P

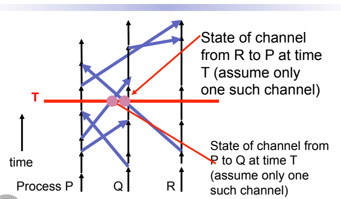
```
while( NOT P) {
   P = detectionAlgorithm();
   sleep(T);
}
// P holds
Q
// P may no longer hold
```

14

GENERAL APPROACH TO DETECTION ALGORITHMS

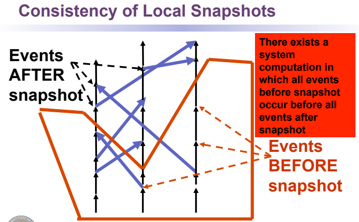
15

System Time Lines: Channel States



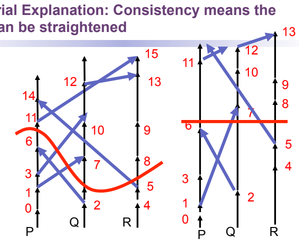
16

Consistency of Local Snapshots

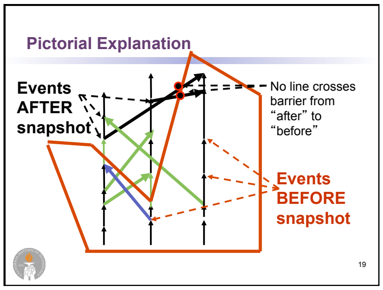


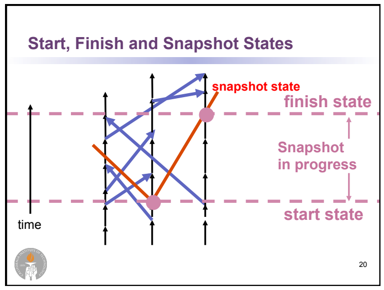
17

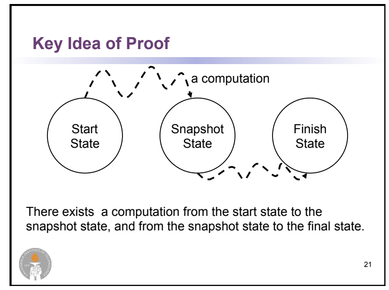
Pictorial Explanation: Consistency means the line can be straightened

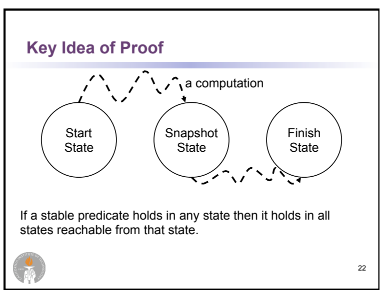


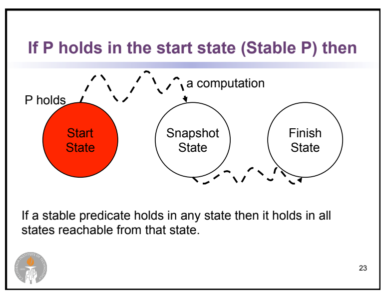
18

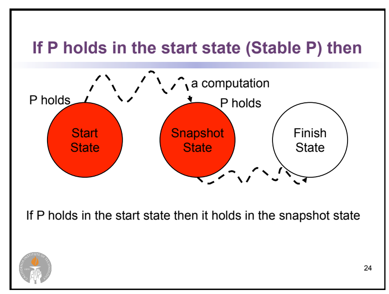

Pictorial Explanation
Events AFTER snapshot
No line crosses barrier from “after” to “before”
Events BEFORE snapshot
19

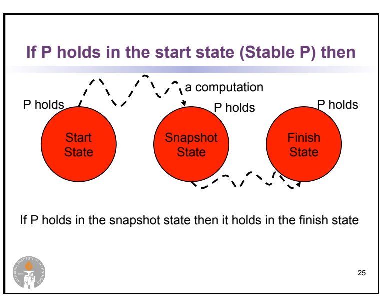

Start, Finish and Snapshot States
snapshot state
finish state
Snapshot in progress
start state
time
20

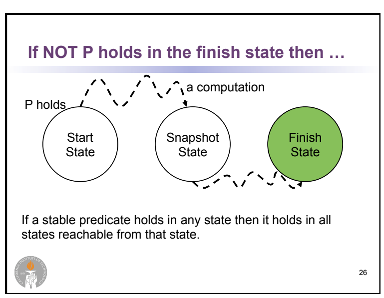

Key Idea of Proof
a computation
Start State
Snapshot State
Finish State
There exists a computation from the start state to the snapshot state, and from the snapshot state to the final state.
21

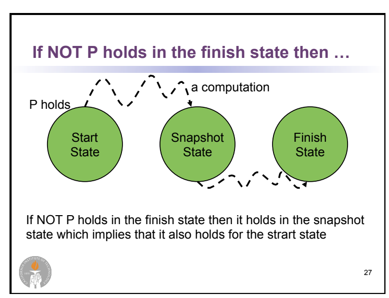

Key Idea of Proof
a computation
Start State
Snapshot State
Finish State
If a stable predicate holds in any state then it holds in all states reachable from that state.
22


If P holds in the start state (Stable P) then
P holds
a computation
Start State
Snapshot State
Finish State
If a stable predicate holds in any state then it holds in all states reachable from that state.
23

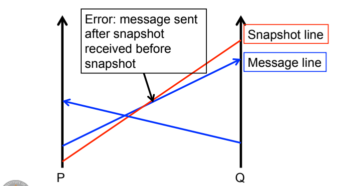

If P holds in the start state (Stable P) then
P holds
a computation
P holds
Start State
Snapshot State
Finish State
If P holds in the start state then it holds in the snapshot state
24


If P holds in the start state (Stable P) then
P holds
a computation
P holds
P holds
Start State
Snapshot State
Finish State
If P holds in the snapshot state then it holds in the finish state
25


If NOT P holds in the finish state then …
P holds
a computation
Start State
Snapshot State
Finish State
If a stable predicate holds in any state then it holds in all states reachable from that state.
26

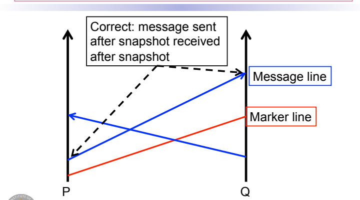

If NOT P holds in the finish state then …
P holds
a computation
Start State
Snapshot State
Finish State
If NOT P holds in the finish state then it holds in the snapshot state which implies that it also holds for the strart state
27

Global Snapshot: Prevents this situation



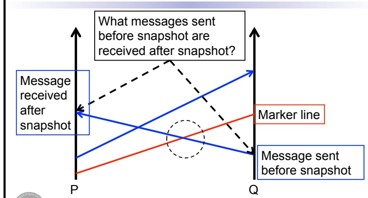
28

Marker message: Prevents this situation



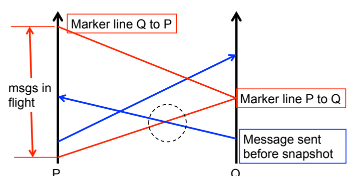
29

Determining messages in flight



30

Determining messages in flight



31

Global Snapshot Algorithm

- 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.

32