FAILURE DETECTORS

Failure De..te..ct..ors

- ALL OUR RECENT PROTOCOLS HAVE DETECTED FAILURES (& THEN RUN LEADER ELECTION/VIEW CHANGE/...)
- NOT THE FOCUS OF TODAY'S TOPIC (DESPITE THE NAME)
- ASYNCHRONOUS — FLP ⇒ ~~Fault Tolerant Consensus~~
  PARTIALLY SYNCHRONOUS — DWORK ET AL ⇒ F.T. Consensus
  SYNCHRONOUS ⇒ F.T. Consensus

SEVERAL WEEKS (& A MIDTERM AGO)

ASYNCHRONOUS — CANNOT DISTINGUISH B/W FAILURE & DELAY

Asynchronous Cannot Distinguish [illegible]

Partially Synchronous

Synchronous Can Distinguish

What Do We Need To Implement F.T. Consensus?

Asynchronous No

?? Yes

Partially Synchronous Yes

How?

Use An Old Idea From TCS: Oracles

↳ Except We Call Them Failure Detectors

Why?

① The Obvious Answers

↳ I chose to inflict this on you

→ Intellectual & Theoretical curiosity

(2) Useful Technique To Evaluate Whether New Hardware Improves Fault Tolerance/Performance/...

→ Does Trusted Execution Improve BFT Consesus?

(Ittai Abrams, Et Al & Suyash Gupta, Et Al)

2022

→ Do Synchronized Clocks Help With Consensus In Fail Stop Model?

(Ghemawat & Lynch, Others)

→ ...

Some Important Points To Remember Before We Start

- Failure Detectors (& Oracles) Usually Cannot Be Implemented
  - ↳ Model To Answer Theoretical Questions (Our Discussion Today)
  - → Model New Hardware That Doesn't Obviously Fit In The FLP Model/ Io Automaton
- Can Sometimes Be Emulated (2nd Paper)
  - ↳ But Not Quite The Same

Back To The Main Topic

First, An Apology About The Paper.

Asynchronous	No F.T. Consensus	No Delay Bounds
??	Yes	??
Partially Synchronous	Yes F.T. Consensus	Eventual Delay Bounds

Problem

- Hard To Reason About Effect Of Delays

- Instead Model Additional Information As Failure Information

  (Notice, This Fits In More Cleanly If We Just Considered Synchronous Vs Asynchronous)

- How To Model Failure Information?
  - ↳ Each Process/Node Can Call A Function F:

$$F : \text{Time} \longrightarrow \text{Set Of 'Suspected' Processes}$$

$$T \longrightarrow 2^{\Pi}$$

- Best Case $F(T)$ Returns Set Of Failed Processes At

TIME T

↳ PHYSICALLY IMPOSSIBLE. WHY?

CHARACTERIZING FAILURE DETECTORS

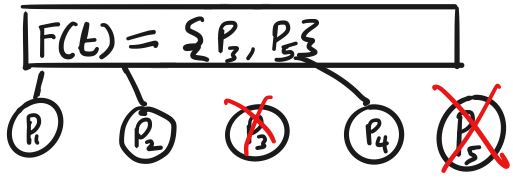
COMPLETENESS: EVENTUALLY HOW MANY FAILED NODES ARE RETURNED

ACCURACY: ARE ANY CORRECT NODES RETURNED

COMPLETENESS

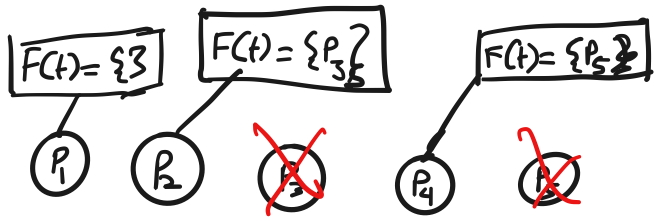
- STRONG: EVENTUALLY ALL CORRECT NODES SUSPECT ALL FAULTY NODES

$\exists t$:



- WEAK: EVENTUALLY SOME CORRECT NODE SUSPECTS EACH FAULTY NODE

$\exists t$:



ACCURACY

- STRONG: NO CORRECT NODE IS EVER SUSPECTED

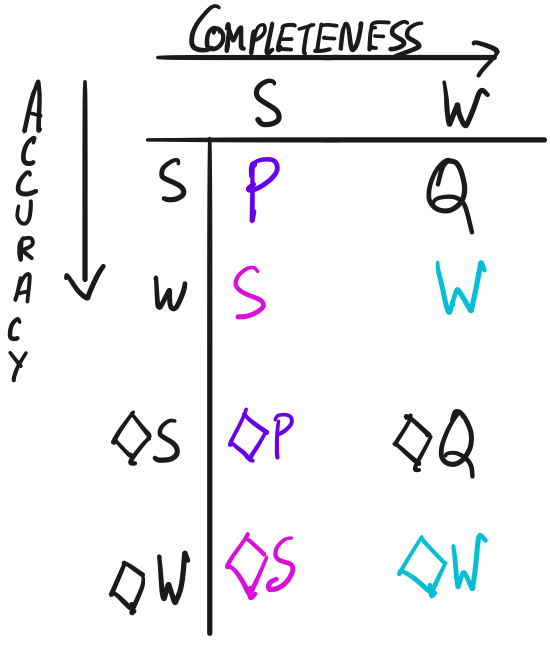
- WEAK: $\exists p \in \Pi$ THAT IS NEVER SUSPECTED

MORE ACCURACY

EVENTUALLY STRONG:

EVENTUALLY WEAK

Putting Them Together

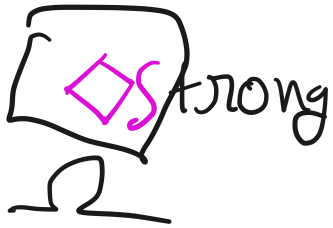


P: Perfect

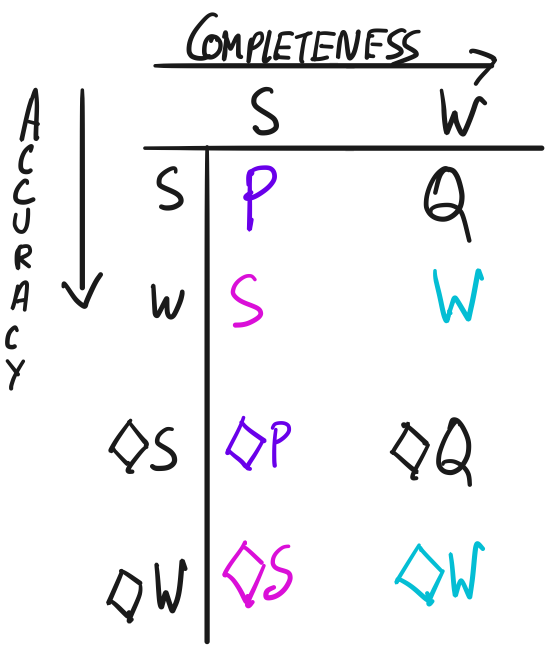
S: "Strong"

W: Weak

Observe: Perfect ⟷ Synchronous



Strong $\longleftrightarrow$ Leader election?



Some relations in Fail Stop

$P \longrightarrow \Diamond P$

$S \longrightarrow \Diamond S$

$W \longrightarrow \Diamond W$

$Q \longrightarrow \Diamond Q$

$P \longrightarrow Q$ $\quad P \longrightarrow S$

$S \longrightarrow W$ $\quad Q \longrightarrow W$

MAYBE SURPRISING

Q → P, W → S

- Strong Completeness: All Correct Processes Suspect All Failed Processes
- Weak Completeness: Some Correct Process Suspects Each Failed Process

→ Both Are Eventual

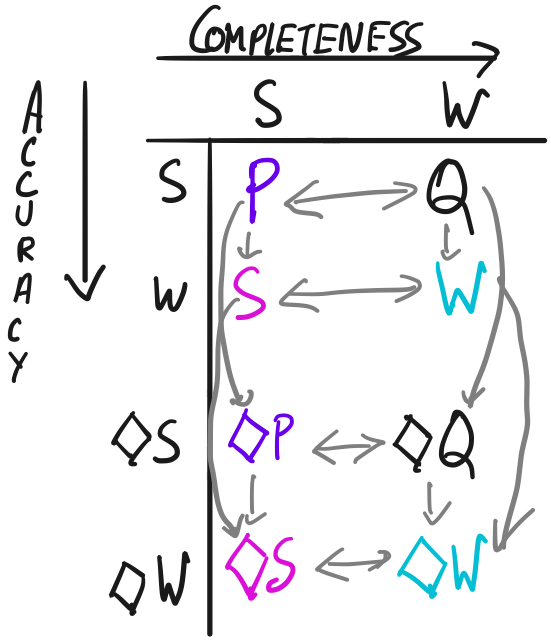
Weak Completeness → Strong Completeness

- Take Union Of FD Output From Different Processes.

How?

- Each process periodically queries F with W.c
- Broadcasts query output & time
- All processes track last query response from each process
- Output of F' with S.C. is union of last responses.

CLAIM. Does not affect accuracy!



CORE RESULT

$\diamond S \longleftrightarrow$ CONSENSUS

EQUIVALENTLY $\diamond S$ ($\diamond W$) IS THE WEAKEST FD FOR F.T. CONSENSUS.

$\diamond S \longrightarrow$ CONSENSUS

◇S → Consensus

- Everyone repeatedly queries ◇S, computer Π − ◇S (correct processes)
- Leader election among processes in Π − ◇ [only wait for votes from all correct processes]
- Eventually elect a correct leader who decides on value & broadcasts.

Consensus → ◇S

- Periodically run consensus protocol
  - ↳ Each node tracks who participated
- Use consensus to agree on nodes who participate
- ◇S returns consensus value

What can we actually infer from this?

Building FDs in Practice

Problem: FLP - previous construction might not work

Bigger Problem: $\diamond S$ not useful in many cases

- Correct nodes might be suspected

  $\Rightarrow$ Protocol must deal with suspected nodes reanimating

- Lerners et al solution