Advanced Network Security #1

Introduction
Distributed Algorithms

message passing
  "Internet"
  - Torrents / P2P
  - Routing
  - time synchronisation
  - DNS

| Shared memory
  "Compute server"
  - map/reduce (?)
  - "folding at home"
  - threading / IPC
  - parallel processing |
Concurrency

j = 1
j = 0

thread i < 2
thread j < i

print j

starts right after the
threats start

Question: what will be printed?

2 i < 2 / j < i / print j
1 j < i / print j / i < 2
0: print j

j < i i < 2
Concurrency

\( i = 1 \)
\( j = 0 \)

Thread: \( i \leftarrow 2 \); print \( j \)
Thread: \( j \leftarrow i \); print \( j \)

Q: What will be printed?

0 2 i \leftarrow \text{print } j; j \leftarrow \text{print } i

1 1 j \leftarrow \text{print } j; i \leftarrow z \text{; print } j

2 2 i \leftarrow z \text{; print } j; j \leftarrow i \text{; print } j

\text{not atomic}

0 1 i \leftarrow z \text{; print } j
\text{reads } i

\text{writes } j \leftarrow 1
\text{prints } j \leftarrow 1
Infinite executions?

\[ i = \phi \]

thread while \( i = \phi \)
do\( \uparrow \) print \( i \)
+ print \( i \)
thread \( i < 1 \)

Q: output printed?

\[ \overbrace{0 \ldots 0 1} \leq \]

\[ 0 \ldots 0 11 \leq \]

\[ 0 \quad \text{or more} \]

\[ \ldots \quad \text{or more} \]

\[ \ldots \quad \text{because of lifeness} \]

Scheduler

Liveness condition:

any enabled action will eventually be executed
non-deterministic

\[ i = 0 \]

\[ \text{thread while } i = 0 \]
\[ \text{do print } i \]
\[ \text{print } i \]

\[ \text{thread } i < 1 \]

\[ \Rightarrow \text{infinite execution is impossible because of liveness} \]

\[ \begin{array}{c}
\{0000111\} \\
p = ?
\end{array} \]

randomness

\[ i = 0 \]

\[ \text{thread while } i = 0 \]
\[ \text{do } i \in \mathbb{R} \{0,1\} \]
\[ \text{print } i \]

\[ \Rightarrow \text{the probability of an infinite execution} = 0 \]

\[ \begin{array}{c}
\{0000111\} : p = (\frac{1}{2})^4
\end{array} \]
Modelling a distributed system

- node
  - execute a sequence of actions: program
  - each of these actions, when executed, is an event
  - communicate with other nodes:
    - shared memory
    - message passing

- Graph $G = (V, E)$
  - $V$: nodes
  - $E$: edges
    - $(u, v) \in E$ if $u$ can send a message (write a register) to $v$

- $G$ directed / undirected
- $|V| = n$
Causality

- let A of events, and let a, b ∈ A events
  - "happens before" relationship: a → b
    - "has an influence on"
      - a, b happen on the same node, and a was first

message passing - a is a send event, and b receives the value sent by a

shared memory - a is a write event, and b reads the value written by a

- a → b ∧ b → c ⇒ a → c
- a → a
An execution of a distribution system is modelled by \( \langle A, \Rightarrow \rangle \) extend into a total order \( \Rightarrow \).