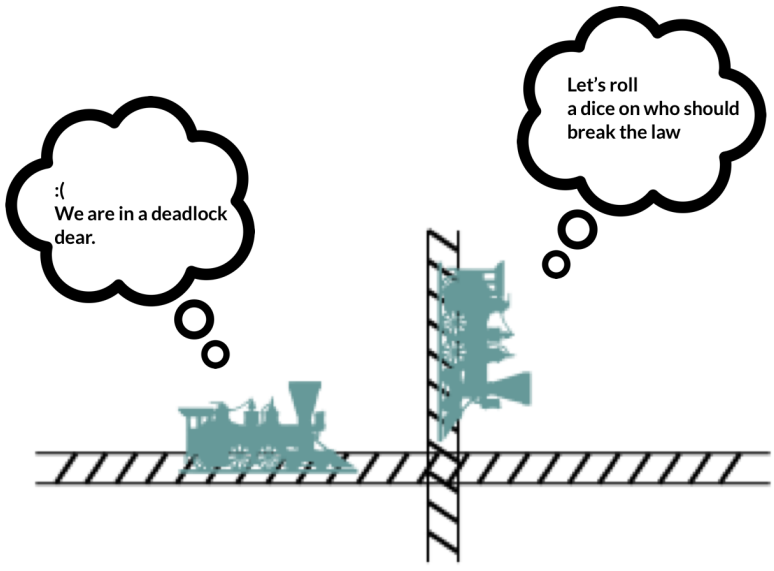
# CSC369 Week 10 Notes

Hyungmo Gu

May 31, 2020

- Deadlock Defined
  - **Google Definition:** Is a situation one typically involving opposing parties, in which no progress can be made.
  - Is permanent
  - Happens to set of processes that
    - Compete for same system resources
    - Communicate with each other
- Example of Deadlock
  - Law passed by Kansas Legislature in in early 20th century
    - "When two trains approach each other at a crossing, both shall come to a full stop and neither shall start upon again until the other is gone"

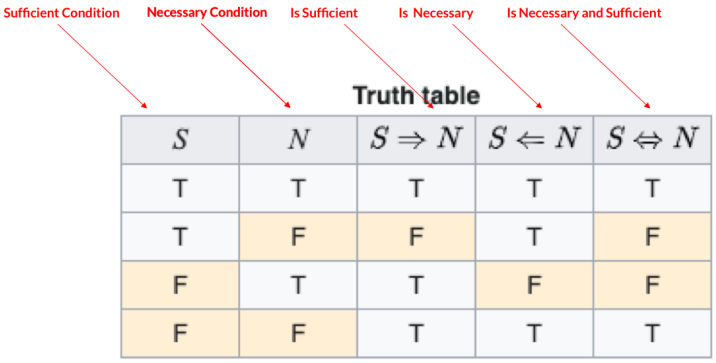



- Conditions for Deadlock

- Neccesary and Sufficient Conditions
  1. Mutual Exclusion
     * Only one process may use a resource at a time
  2. Hold and wait
     * A process may hold allocated resources while awaitng assignment of others
  3. No preemption
     * No Resource can be forcibly removed from a process holding it
  4. Circular wait
     * Each process must be waiting for a resource which is being held by another process, which in turn is waiting for the first process to release the resource [3]

**Aside**

1. Wait. Necessary condition? [1]
   - We say $N$ is a necessary condition for $S$ if we don't have $N$, we won't have $S$.
2. Wait. Sufficient condition? [1]
   - We say $S$ is a necessary condition for $N$ if we have $S$, then we know that $N$ must follow, i.e. $S \Rightarrow N$
3. Hold on. How about necessary and sufficient condition?
   - Is when necessary and sufficient conditions are put together similar to if and only if [2]




| $S$ | $N$ | $S \Rightarrow N$ | $S \Leftarrow N$ | $S \Leftrightarrow N$ |
|---|---|---|---|---|
| T | T | T | T | T |
| T | F | F | T | F |
| F | T | T | F | F |
| F | F | T | T | T |

**References**

1) Fayetteville State University: Necessary and Sufficient Conditions, link

2) Wikipedia: Necessity and Sufficiency, link
3) Wikipedia: Deadlock, link

- Solutions
  - Prevention
    - Ensures that at least one of the necessary conditions to deadlock will never occur [1]
  - Avoidance
    - Ensures that the system will not enter an unsafe mode [1]
  - Detection and Recovery
    - Ensures that the system recovers from deadlock if a deadlock has occured in the system. [2]
  - Do Nothing and hope
    - Is done by Windows, Linux, and JVM
    - Works until eventually the deadlock snowballs, no longer functions, and requires manual intervention (Press power, and restart) [3]

**References**

1) pediaa: What is the Difference Between Deadlock Prevention and Deadlock Avoidance, link
2) Geeks for Geeks: Recovery from Deadlock in Operating System, link

- Safe States
- Unsafe States & Algorithm
- What is Atomicity?
- Why would atomicity fail?
- Definitions for Transactions
- How to ensure atomicity in the face of failures?
- Write-ahead logging
- Problems with logging
- Deadlock and Starvation
- Communication Deadlocks
- Livelock