Repairable systems

Repairable system

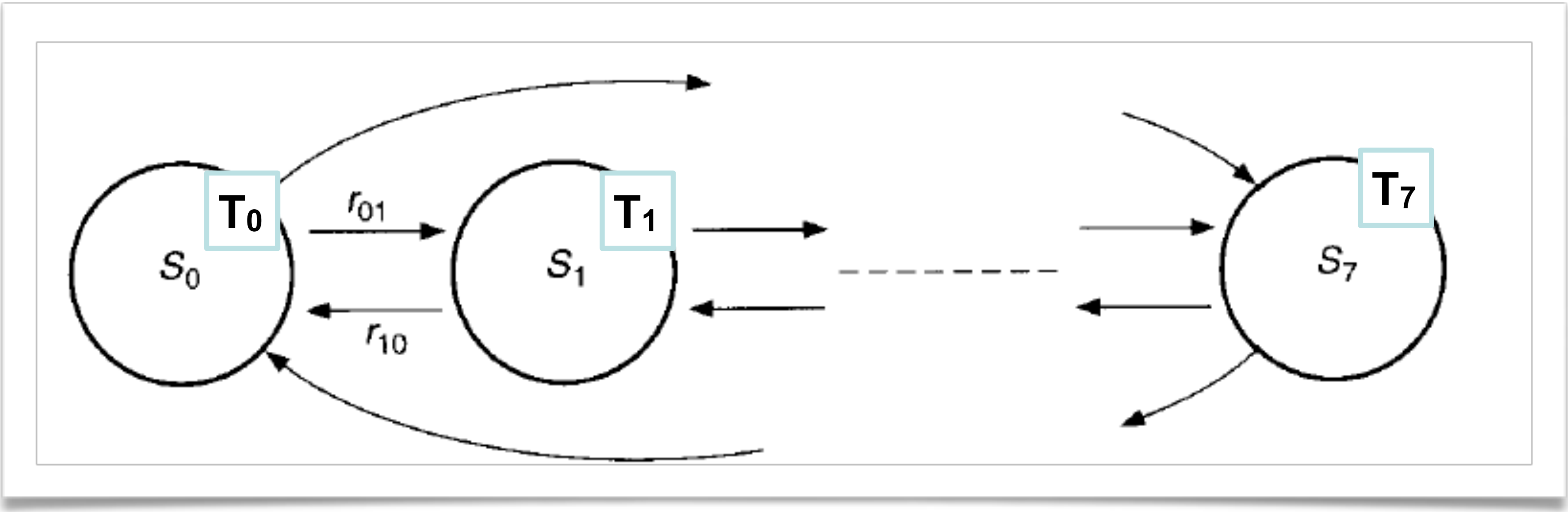
- A reparable system is obtained by glueing individual non-repairable systems each around a single failure

- To describe this gluing process we need to review the concept of stochastic process

Stochastic process

- Stochastic processes are classified by their
  - **State Space** – the range of possible values
  - The **index set** of the state space
  - The **dependence structure** among random variables X_t that make up the whole stochastic process

Stochastic process



Failure occurrences

- For each index i we have a new random variable Ti representing the Time of the i-th Failure
- Each Ti has its own probability density function fi, cumulative distribution function Fi and hazard rate hi

Failure occurrences

- We introduce a new random variable which is the **Time Between Failures**

$$X_1=T_1 \text{ and } X_i=T_i-T_{i-1}$$

- Note: giving Time of Failure you can easily derive the Time Between Failures and vice versa

Failure occurrences

- Each Time Between Failures X_i (or T_i) is a random variable for each state of the process
- These variables may be
  - Dependent or independent
  - Identically distributed or not identically distributed

Independent random variables

- Two random variables X and Y say, are said to be **independent** if and only if the value of X has no influence on the value of Y and vice versa

Identically distributed random variables

- Two random variables X and Y are said **identically distributed** if they have the same cumulative distribution function F (or density function)

Repairable systems

- The assumption of independent and identically distributed **times between failures** is usually invalid for software repairable systems

- Why?

Repairable systems

- In classical **hardware theory**, we simply **replace** failed components with **identical working** new ones
  - We might have that **all density functions** to be **identical** and their cumulative distribution function as well

- Generally speaking: substituting a part of a car does not change the car performance (density function) because a mechanic cannot intervene on its design!
- In few cases, a mechanic may also replace a failed component with one of better quality though

Repairable systems

- Once a software fault is completely removed it will **not cause the same failure again**, but ...
- **Dependency**: Removing faults may cause new failures: the variable **Times Between Failures X_i may be dependent**

- By fixing a failure we may also **improve the design** to minimize the likelihood of recurrence of the faults that have caused the failure
- Fault **prevention:** in operation settings, software reliability can also be improved by **testing** whereas for hardware one has to use better material

Repairable systems

- Altogether, we expect that the probability density function of X_i would be **different** than the one of X_{i-1}
    - For example, by improving design $E[X_{i-1}]$ tends to be less than the one of $E[X_i]$

Minimal/Perfect repair

- **Minimal repair (as bad as old):** the repair done on a system leaves the system in exactly the same condition as it was just before the failure
- **Perfect repair (as good as new):** the system is brought to a new state after the repair

- If every repair is a **perfect repair** then times between failures are independent and identically distributed