Introduction to Modeling

1. What is a model?
2. Modeling Methodologies
3. Modeling a Computer

1. What is a Model?

- Model : a representation that mimics another object under study.
  - Physical, logical, functional representation.
  - Modeling : 모델화 하는 것. 시스템을 모형화 하여 분석하여 우리가 시스템에 대해 알고자 하는 사항들을 도출해 내는 것
- The particular features of the model that are implemented depends on the user's requirements and desires.
  - Ex : a toy airplane, a model house, 건축물 모형, ...
  - The particular features of the model are represented.
  - Simple, easier, cheaper to understand the real object

❑ Kinds of models

1. Physical model
   - Physical outlook of an object
   - Model is constructed out of plastic, wood, metal, ...
   - Its operation follows the same mechanisms
   - 예) toy airplane, 의상 모델, 건축물 모형

2. Simulation model
   - Mimics real behavior of the object
   - Manipulation of values that represent physical observation
   - 예) war game, 전자오락 자동차 경주, 은행 이용 방식

3. Analytical model
   - Does not represent operations that mimics the behavior of the object
   - Mathematical operations to capture the relationships
   - 예) 은행 이용 방식 중 고객의 대기 시간 분석

2. Modeling Methodologies

- Performance modeling method
  1. measurement and empirical models
  2. simulation(models)
  3. analytic models

- Empirical modeling
  - 장점
    - 정확, 실제적 모델(가정이 거의 없음)
  - 단점
    - 시스템 구현 후 가능
    - 데이터 수집까지 장시간 소요
    - 고 비용
    - 측정된 데이터 자체는 정확하나 포괄적이지 못함

- Simulation modeling
  - 장점
    - 모델 수립이 비교적 용이
    - 시스템 구현 전에 적용 가능
    - 부분적으로 포괄적
    - 동작 및 절차를 표현하는데 적합
  - 단점
    - 시뮬레이션 반복 회수에 따라 정확성 좌우 → 장시간 소요 → expensive
    - 입력 변수 변경 시 완전 재수행
    - 모델 수립 시 세부 동작은 생략 (모델의 부정확)
    - 모델 실행 시 확률적 가정에 의지
    - 시뮬레이션 툴 필요

- Analytic modeling
  - 장점
    - 모델의 입력 변수 변경 시 신속 수행
    - 시스템 구현 전 적용 가능
    - 포괄적
    - 수학적 모델의 해석, 정확성
    - 가장 저렴한 방법
  - 단점
    - 모델 수립의 어려움(수학적 skill)
    - 모델 수립 시 세부 동작은 생략 (모델의 부정확)
    - 모델 수립 시 확률적 가정에 의지

3. Modeling a Computer

- Computer system
  - Consists of many components
  - Complex, but fast operations
  - A large volume of input and output, system states
    - → Needs simplified model to understand
- Level of modeling
  - Overly simplified model for a casual user
  - Complex and detailed model for computer engineers
    - → needs a formal way to describe complex models correctly and efficiently
- examples of formal modeling techniques
  - Formal Description Techniques, Simulation Languages, State Transition Diagram, Mathematical Equations.

- The computer performs many operations in a period of time, so its actions are similar to physical processes.
  - → These physical processes are termed stochastic since they appear randomly as a function of time.

- A stochastic model is different from other models.
  It models systems that will have many possible outputs for a single set of inputs

- What to model for a computer
  - Performance
  - Reliability
  - Operations

Performance Measures

Performance Measures		
Measure	**Units**	**Potential Use**
Throughput	Processes/unit time	Productivity evaluation
Capacity	Processes/unit time	Planning
Response time	Units of time	Usability evaluation
Utilization	Percent of time	Configuration
Reliability	Mean time to fail, Probability	Maintenance scheduling
Availability	Percent of time	Usability evaluation
Speedup	Number of effective PUs	Configuration
Backlog	Number of processes	Usability evaluation

- Workload Parameters

Workload Parameters		
Parameter	**Units**	**Potential Variations**
Interarrival time	Unit of time	Change the offered load
Task size	Unit of time	Amount of processing
I/O request rate	Number per unit time	Type of processing
I/O service rate	Number per unit time	I/O device type
Memory size	Kilobytes per task	Multiprogramming level
Task mix	Number of tasks	Interactive/batch
Parallelism	Percent of program	Vector/sequential

Probability Theory

1. Definition
2. Basic Models
3. Combinatorics
4. Random Variables

1. Definition

- Probability theory : a mathematical model that represents the relative frequency of an outcome for an infinite number of repetitions of the experiment.

$$\text{relative frequency} \triangleq \frac{\text{number of observations of an outcome}}{\text{number of repetitions of the experiment}}$$

- Actual type of measurement for both the inputs and outputs of a process depend on the environment

- Two basic types of environment

  1. Discrete environment
     - Countable outcomes, a finite or countably infinite number of outcomes
     - ex) Coin flipping, dice tossing, number of humans
  2. Continuous environment
     - Uncountably infinite number of possible outcomes
     - ex) Room temperature

2. Basic Models

- Def. 1

  An element is an instance of an object of interest.

- Def. 2

  A set is a collection of distinct elements.

  c.f. : A bag is a collection of elements.

  (예) {1, 2, 3} {m|m=2n, n= 1, 2, ····} : even number

  {{1,2}, 3} { } : null set

  { 0, 1, 2, 3, ····} countably infinite number of elements

  { x | x∈ [0,1]} uncountably infinite number of elements

Def. 3

The union of two sets A and B contains all elements found in either of the two sets.

(예) A={1, 2}, B={3, 4, 5} $A \cup B=\{1, 2, 3, 4, 5\}$

Def. 4

The intersection of two sets A and B contains only elements that appear in both sets.

(예) A={1, 2, 3}, B={3, 4} $A \cap B=\{3\}$

Def. 5

A set A is a subset of a set B if all the elements in the set A are also in the set B.

(예) A={1, 2, 3, 4, 5}, B={3, 4} $A \supset B$

Def. 6

The complement of a set is the set of all elements in the universe(모집단, Ω,가능한 모든 element 로 된 집합) that are not in the set.

(예) $\Omega = \{1, 2, 3, 4, 5\}$ $A = \{1, 2\}$, $\overline{A} = \{3, 4, 5\}$

Def. 7

Two sets A and B are mutually exclusive (disjoint)
if they have no elements in common ($A \cap B = \emptyset$).

Def. 8

Two sets A and B are mutually exhaustive if $A \cup B = \Omega$.

Def. 9

Two sets A and B partition the universal set Ω
if they are both mutually exclusive ($A \cap B = \emptyset$)
and mutually exhaustive ($A \cup B = \Omega$).

Laws of Set Theory

Laws of Set Theory	
1. Commutative	$A \cap B = B \cap A$
	$A \cup B = B \cup A$
2. Associative	$A \cap (B \cap C) = (A \cap B) \cap C$
	$A \cup (B \cup C) = (A \cup B) \cup C$
3. Distributive	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
4. Identity	$A \cap \Omega = A$
	$A \cup \emptyset = A$
5. Inverse	$(A')' = A$
6. DeMorgan's	$(A \cap B)' = A' \cup B'$
	$(A \cup B)' = A' \cap B'$
7. Exclusion	$A \cap A' = \emptyset$

Table 2.1

Def. 10

The sample space for an experiment is the set of all possible outcomes.
(예) Example of flipping a coin, S={H, T}

Def. 11

An event is a set of outcome. (subset of the sample space)
(예) S={HH, HT, TH, TT}
events : {HH, HT, TH, TT}, {HT, TH}, {HH}

Laws of Probability

- Def 12

  For an sample space Ω, a probability measure P is a function defined on all the subsets of Ω such that :

  1. $P[\Omega]=1$
  2. For any event $A \subset \Omega$, then $P[A] \geq 0$
  3. For any events $A, B \subset \Omega$ where $A \cap B = \emptyset$, then
     $P[A \cup B] = P[A] + P[B]$
  3'. For mutually disjoint events *Am*,

$$P\left[\bigcup_{m=1}^{\infty} A_m\right] = \sum_{m=1}^{\infty} P[A_m]$$

(예) For A, A′ such that $\Omega = A' \cup A$ and, $A' \cap A = \emptyset$,

$P[\Omega] = P[A' \cup A] = P[A'] + P[A] = 1$

$\therefore \; P[A'] = 1 - P[A]$

(예) Probability measures are functions that give real values between 0 and 1.

❑ Conditional Probability

A situation : A model of a process has been established with its sample space and its probability measure.

→ The sample space is too general.

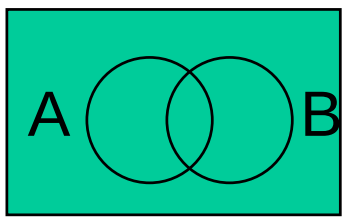
→ A condition should be added to the model to restrict the possible outcomes.

- Def. 13

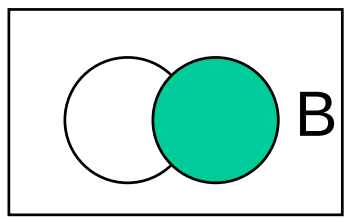
  The conditional probability P[A|B] is the probability measure for an event A given that only outcomes which are in event B are considered, i.e. the probability of event A given event B

- Evaluation of P[A|B]

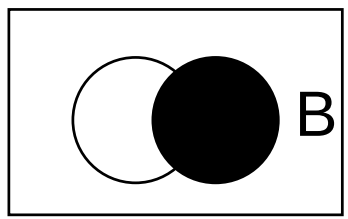
P[Ω]=1



P[B| Ω]=P[B]



P[B|B]=1



- P[B|B] : "B event가 발생했다는 가정 하에"

  → Sample space를 Ω에서 B로 조정"scale down"

  → 문제의 범위를 B로 제한

  "B가 일어날 확률"

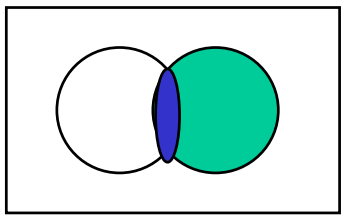
- P[A|B] : "B가 발생했다는 가정하에 outcome 이 A에 속할 확률"

  "B가 발생했다는 가정하에 A가 발생할 확률"

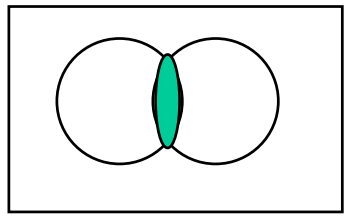
  "B범위 안에서 A가 발생할 확률"

  "outcome 이 B에 속한다고 가정하고 그것이 A에 속할 확률"

P[A|B]



P[A∩B]



P[A∩B] : “outcome이 B에 속할 확률 그리고 동시에 A에도 속할 확률”

$$P[A|B] = \frac{P[A \cap B]}{P[B]} \quad , \quad P[B] \neq 0$$

P[A∩B], P[B] : old probability measures of Ω

◆ (예) 동전 2개 던지기

Ω = {HH, HT, TH, TT}

event A=동전 두 면이 모두 H이다. S_1={HH}

event B=동전 두 면 중 최소한 한 개는 H이다. S_2={HH, HT, TH}

$$P[A] = \frac{1}{4} \qquad \{HH\} / \{HH, HT, TH, TT\}$$

$$P[A|B] = \frac{1}{3} \qquad \{HH\} / \{HH, HT, TH\}$$

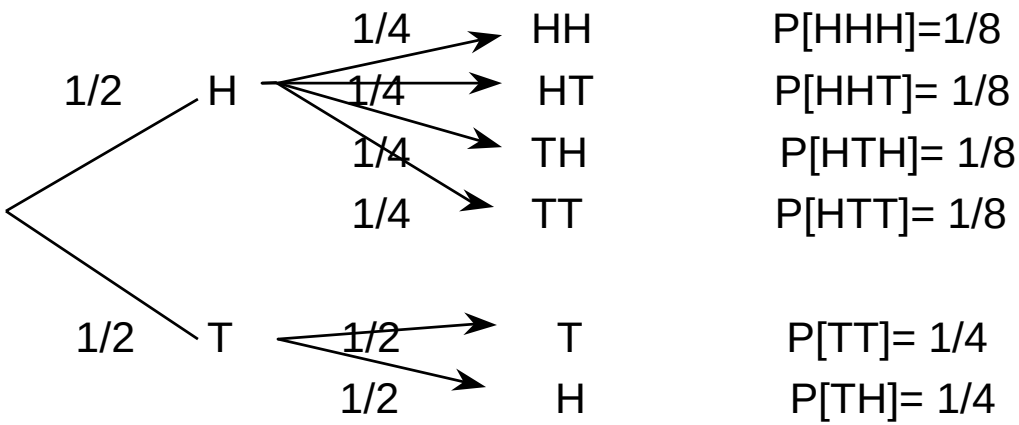
$$= \frac{P[A \cap B]}{P[B]} = \frac{P[A]}{P[B]} \qquad (A \subset B)$$

$$= \frac{\frac{1}{4}}{\frac{3}{4}}$$

Inversion of the Formula $P[A \cap B]=P[B] \cdot [A|B]$

- P[A|B] 와 P[B]를 가지고 P[A∩B]구하기
- Since [A|B] is a restricted case,
  it is often easier to determine that value than P[A∩B]

- (예) 동전 던지기, 한번 던져서 H가 나오면 동전 2개를 더 던질 수 있고, T가 나오면 동전 1개를 추가로 던진다. 동전을 최대 3개를 던져 연속된 T를 얻을 확률
  Probability Tree



A : 연속된 T를 얻는 event, B : 동전 3개를 던지는 event (첫 동전이 H일 event)

Independence

- Def. 14

  Two events A, B $\subset \Omega$ are independent if and only if $P[A \cap B] = P[A] \cdot P[B]$.

  - The probability of an event does not change when a condition is given.

    $$P[A|B] = P[A] = \frac{P[A \cap B]}{P[B]}$$

  - Intersection is reflexive ($A \cap B = B \cap A$), so A being independent of B means B being independent of A.

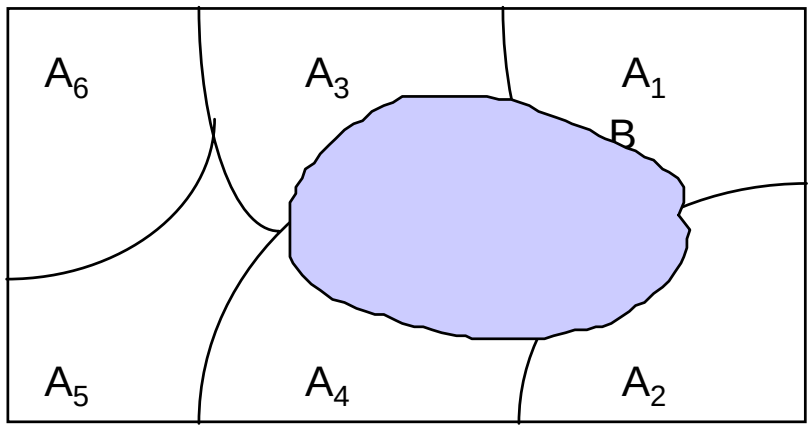
Bayes' Theorem

Bayes' Theorem

In a set of mutually exhaustive and exclusive events which form the partition of the sample space and the fact that the event of interest has nonzero probability.

$$P[A_i \mid B] = \frac{P[A_i \cap B]}{P[B]} = \frac{P[A_i]P[B \mid A_i]}{\sum_j P[A_j]P[B \mid A_j]}$$

Condition domain을 바꿈으로써 문제를 간단히 해석 가능



◆ 예) 세 사람의 프로그래머가 프로그램을 짜서 실행시키는데 실행시킬 때 마다 성공하거나 실패의 결과를 얻음.
다음과 같은 확률을 알고 있을 때(모두 쉽게 측정가능)

Event A_1 : 어떤 pgm을 프로그래머 1이 구현했음. $P[A_1]=0.2$
Event A_2 : 어떤 pgm을 프로그래머 2이 구현했음. $P[A_2]=0.3$
Event A_3 : 어떤 pgm을 프로그래머 3이 구현했음. $P[A_3]=0.5$
Event B : 구현된 프로그램이 실패함. $P[B|A_1]=0.1$
$P[B|A_2]=0.7$
$P[B|A_3]=0.1$

어떤 실패한 프로그램이 프로그래머 1에 의해 구현된 것일 확률은?

$$P[A_1|B] = \frac{P[A_1]P[B|A_1]}{\sum_j P[A_j]P[B|A_j]}$$

$$= \frac{0.2 \times 0.1}{0.2 \times 0.1 + 0.3 \times 0.7 + 0.5 \times 0.1}$$

$$= 0.07144$$

3. Combinatorics

N : A number of distinct elements

- Number of different samples of length R with replacement

  $= N^R$

- Number of different samples of length R without replacement

  $$= \frac{N!}{(N-R)!}$$

- Number of possible arrangement by choosing R out of N

  : Permutation of N taken R at a time

  - P(N,R)
  - P(N,N) =1, P(N,0) =1

- Number of combinations of N things taken R at a time

$$= \frac{N!}{R!\cdot(N-R)!} \triangleq \binom{N}{R}$$

- Number of subsets of size R we can obtain from the set of size N

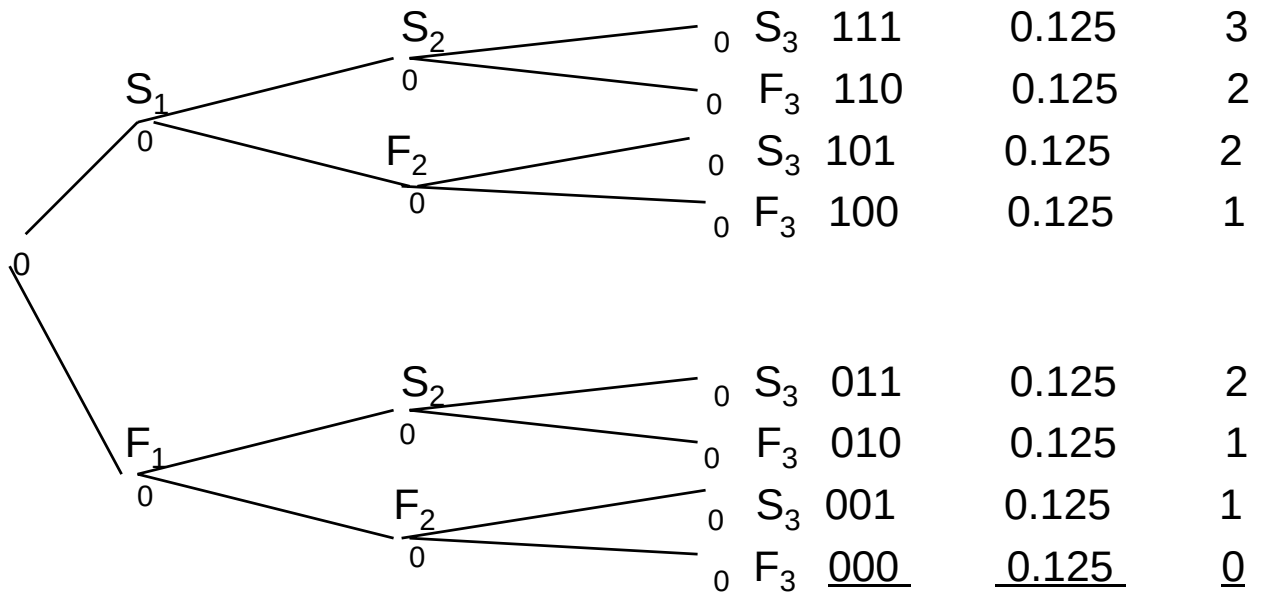
$$\sum_{R=0}^{N} \binom{N}{R} = 2^{N}$$

4. Random Variables

Def. 15

A random variable is a function that assigns a value(real number) to each possible outcome in the sample space.

- 예) 주사위 게임 짝수 : win, 홀수 : lose
  win $\rightarrow$ 0 lose $\rightarrow$ 1
- 예) 기준 온도에서 증/감 분량 표시하는 온도계
  매 분마다 측정 후 증/감 온도 $\rightarrow$ 현재 온도에 대한 function
- 예) Consider a random experiment defined by a sequence of three Bernoulli trials.
  - The sample space S consists of eight triples of 0's and 1's.
  - Define a random variable X to be the total number of successes from the three trials.
  - The tree diagram : S_n and F_n represent a success and a failure respectively on the n-th trial
  - The probability of success, *p*, is equal to 0.5.
  - The value of random variable X assigned to each sample point is also shown.


S1
F1
S2
F2
S2
F2
S3 111 0.125 3
F3 110 0.125 2
S3 101 0.125 2
F3 100 0.125 1
S3 011 0.125 2
F3 010 0.125 1
S3 001 0.125 1
F3 000 0.125 0
sample points(s) P(s) X(s)

Def. 16

The probability mass function (pmf) of the random var. X is a function with the domain consisting of the event space of the random var. X, and with its range in the closed interval [0. 1]

$$P_X(x) = P[X = x] = \sum_{X(s)=x} P(s)$$

Properties of the pmf

Properties of the pmf

(1) $0 \le P_X(x) \le 1$: $P_x(x)$ is a probability

(2) Since the random variable assigns some value x∈R to each sample point s∈S, we have

$$\sum_{X \in R} P_X(x) = 1$$

(3) For a discrete random variable X such that

{x_1, x_2, x_3,····}

$$\sum_{i} P_X(x_i) = 1$$

예) 직전 동전 3개로의 Bernoulli 실험

For x=0, 1, 2, 3 Px(0) = 0.125

Px(1) = 0.375

Px(2) = 0.375

Px(3) = 0.125

❑ Cumulative Distribution Function(CDF)

- CDF 또는 Probability distribution function, distribution function
- Probability mass function Px(x)는 random variable X 가 x의 값을 가질 때의 함수
- 어느 set {s|X(s) ∈A} for some subset A of R의 확률을 구하고자 할 때 CDF이용
- P[X∈A]는 event[X∈A]의 확률

  P[X∈(a,b)] --> P(a < x < b)

  P[X∈(a,b]] --> P(a < x ≤b)

$$\{s \mid X(s) \in A\} = \bigcup_{x_i \in A} \{s \mid X(s) = x_i\}$$

$$P[X \in A] = \sum_{x_i \in A} P_X(x_i)$$

- 예) 동전 3개로 Bernoulli 실험 예

  P[X≤1] = P[X=0] + P[X=1]

  = Px(0) + Px(1) = 0.125 + 0.375 = 0.5

- Def. 17

The function Fx(t), $-\infty < t < \infty$, defined by

$$F_X(t) = P[-\infty < X \le t] = P[X \le t] = \sum_{x \le t} P_X(x)$$

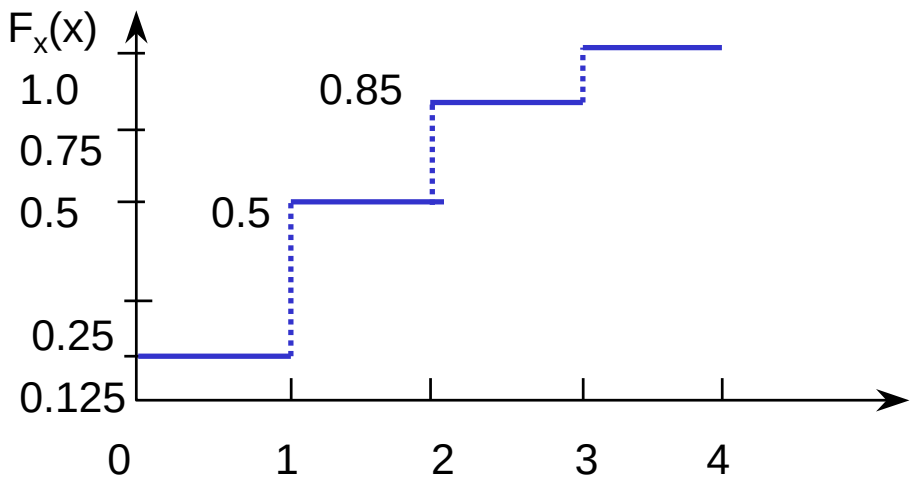
is called cumulative distribution function(CDF) or the probability distribution function of the random variable X.

$$P[a < X \le b] = P[X \le b] - P[X \le a] = F[b] - F[a]$$

If X is an integer value random variable, then

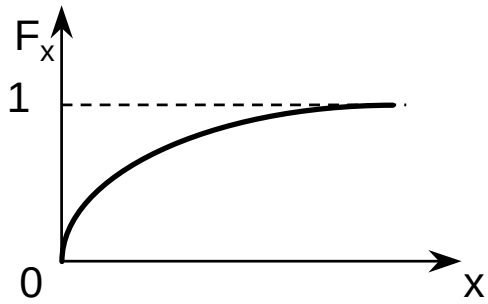
$$F(t) = \sum_{-\infty < x \le \lfloor t \rfloor} P_X(x)$$

◆ 예) Bernoulli 실험 예의 CDF



$$F(0) = P[-\infty < X \le 0] = P[X \le 0] = 0.125$$
$$F(1) = P[X \le 1] = P[X = 0] + P[X = 1] = 0.5$$
$$F(2) = P[X \le 2] = F(1) + P[X = 2] = 0.875$$
$$F(3) = P[X \le 3] = 1$$

* 연속적 r.v. 의 경우



- Three properties of CDF

$$(1)\ F(-\infty) = 0$$

$$(2)\ F(\infty) = 1$$

$$(3)\ \text{For } x_1 < x_2, \quad \textit{then } F(x_1) \le F(x_2), \quad -> F(x) \text{ is nondecreasing}$$

$$==>\ 0 \le F(x) \le 1$$

- 4가지 대표적 CDF
  - ① Geometric distribution : infinite set, discrete
  - ② Exponential distribution : infinite set, continuous
  - ③ Poisson distribution : infinite set, discrete
  - ④ Binomial distribution : *finite* set, discrete of possible values

Geometric Distribution

Geometric Distribution

- Head 나올 확률 1-p, tail 나올 확률이 p인 동전
- Let K be the random variable that represents the number of flips required to obtain the first head

P[flip the first head on k-th flip]
= (1-p) p^{k-1} for k = 1, 2, ····

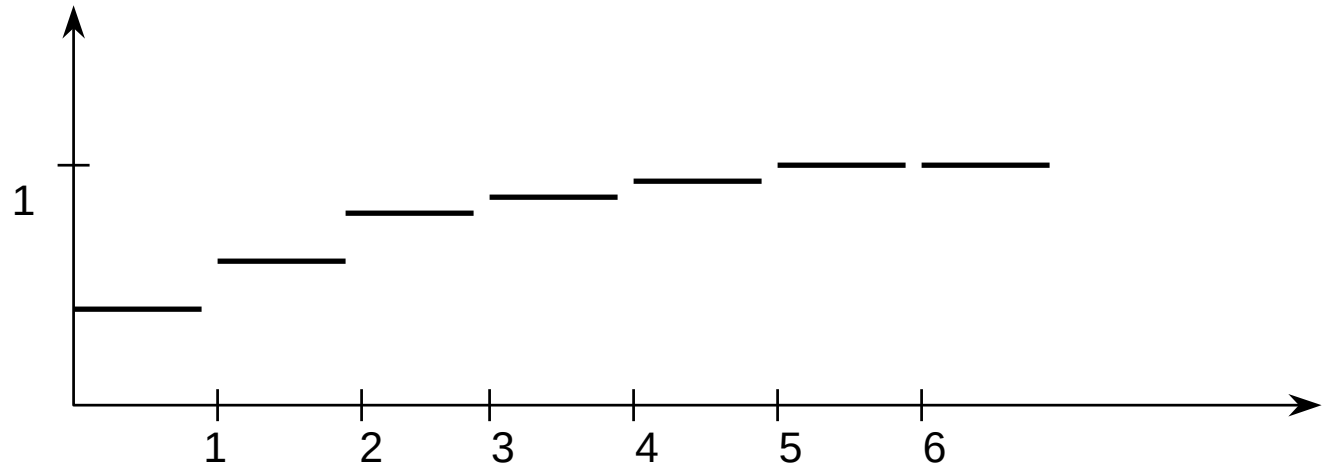
$$F(k) = \sum_{n=1}^{k} P[\text{flip the first heads on n - th flip}]$$

$$= \sum_{n=1}^{k} (1-p)\, p^{n-1} = 1 - p^{k}$$

Modified Geometric pmf : the random variable that represents the number of flips before we obtain a head,

P[flip the first head on (k+1)st flip] = (1-p) p^{k} for k = 0, 1, 2, ····

A plot of the modified geometric distribution



Binomial Cumulative Distribution

Binomial Cumulative Distribution

- Random variable K is the number of heads appearing in those N coin flips

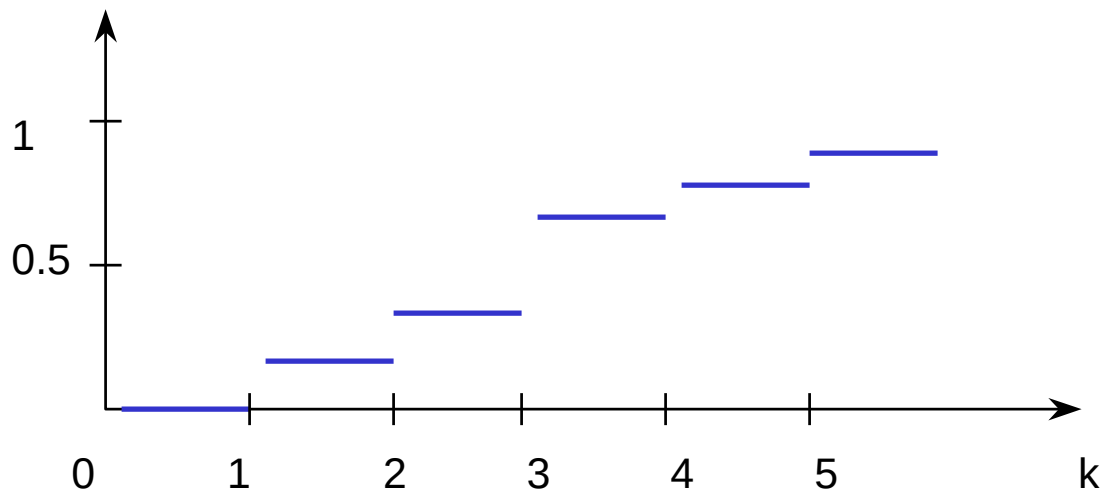
  P[a sequence of N flips with k heads] = $(1-p)^{k}p^{N-k}$

- Many different sequences with k heads

$$P[k \text{ heads}] = \binom{N}{k}(1-p)^{k}(p)^{N-k}$$

- CDF

$$F(k) = \sum_{m=0}^{k} \binom{N}{m} (1-p)^{m} p^{N-m}$$

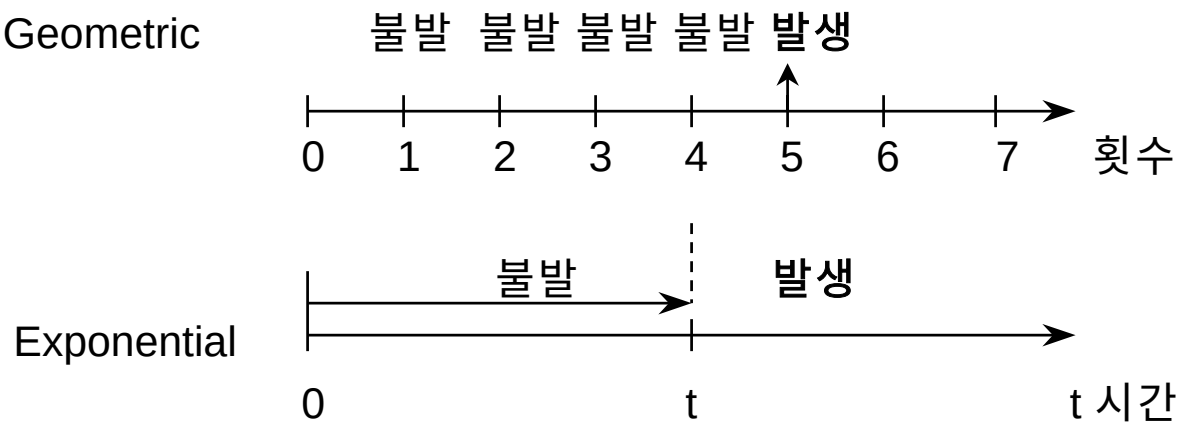


Exponential Cumulative Distribution

Exponential Cumulative Distribution

- continuous version of the geometric distribution
- random variable T represents the amount of time until a success occurs.

- 예) Geometric



Success happens at some rate λ

$$\text{rate } \lambda = \frac{\text{평균발생횟수}}{\text{단위시간}} \qquad \text{예) } \lambda = 0.1$$

$$\frac{1}{\lambda} = \frac{\text{단위시간}}{\text{평균발생횟수}} = 1\text{회 발생 때 까지의 평균시간}$$

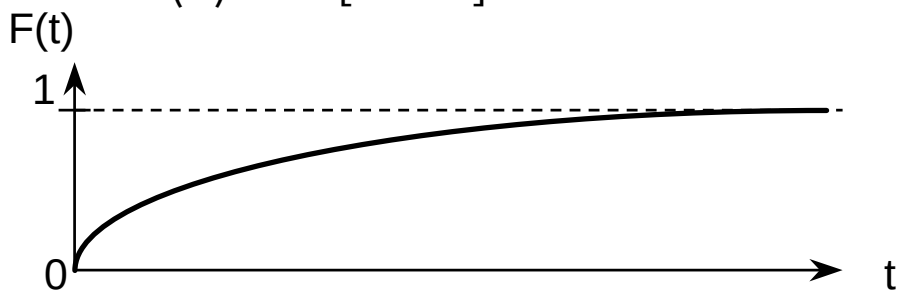
예) Hard disk 고장율 $\lambda = 0.0001$ times/hour

1시간에 평균 0.0001번 고장

1회 고장까지 평균 10000시간 소요

- CDF

$$F(t) = P[T \le t] = 1 - e^{-\lambda t}$$



- success의 예 : arrival of a job, completion of a task, response by a user

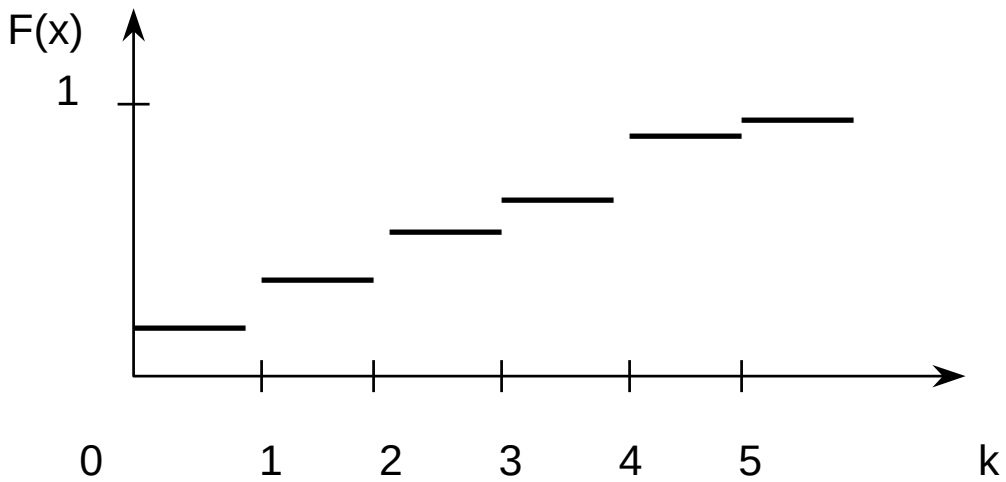
Poisson Cumulative Distribution

- Poisson Cumulative Distribution
  - Continuous version of the binomial distribution
  - Trials are allowed at any moment in time
  - Random variable k of successes during a fixed period of time T and a success rate λ

$$P[\text{k successes in time T}] = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

- CDF

$$F(k) = \sum_{n=0}^{k} \frac{(\lambda T)^n}{n!} e^{-\lambda T}$$



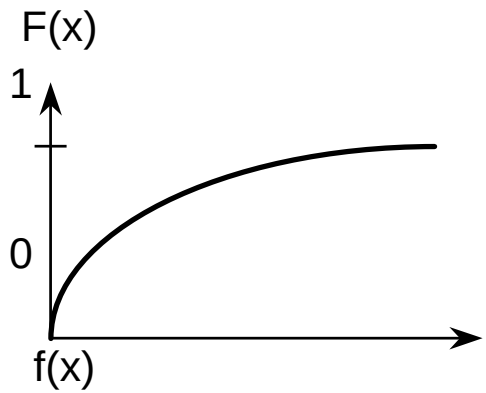
Probability Density Functions

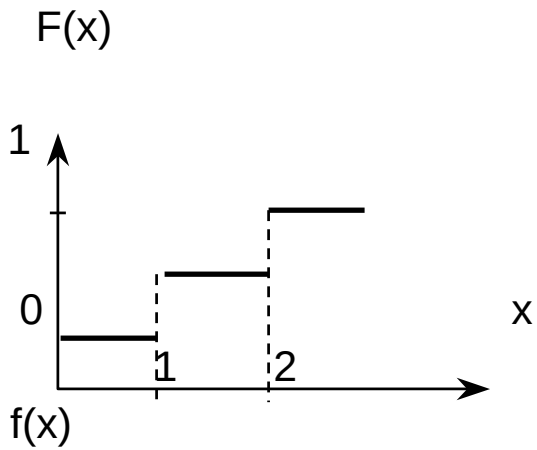
- 특성
  - Continuous version of probability mass function(pmf)
    - ↳ Functional representation of the probability measure
  - The density function is a measure of the amount of the increase in the cumulative distribution func.(pmf 에서와 같이 probability measure는 아님)

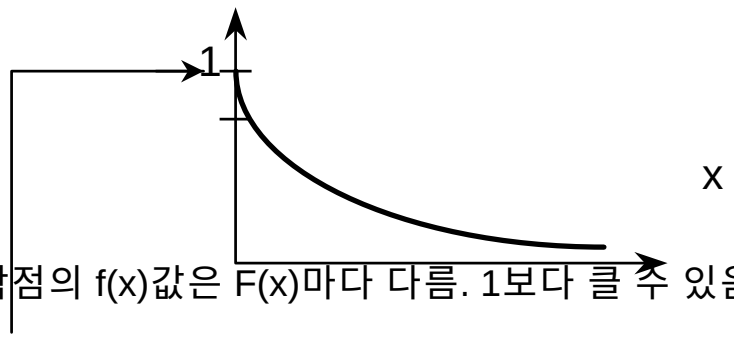
- Def. 18
  - The probability density function f(x) is the derivative of the cumulative distribution function F(x) :

$$f(x) \triangleq \frac{d}{dx} F(x) \longrightarrow F(x) = \int_{-\infty}^{x} f(y)dy$$

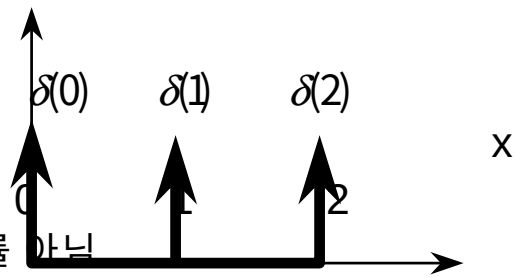
- Define the derivative of the distribution function at a discontinuous jump to be a impulse function $\delta(x)$







시작점의 f(x)값은 F(x)마다 다름. 1보다 클 수 있음. f(x)는 확률 아님



Basic properties of the Density Function

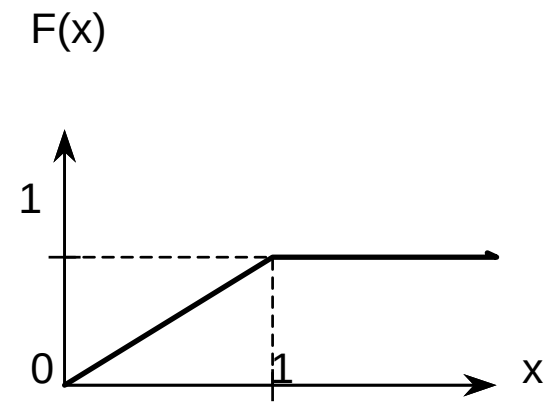
- Basic properties of the Density Function

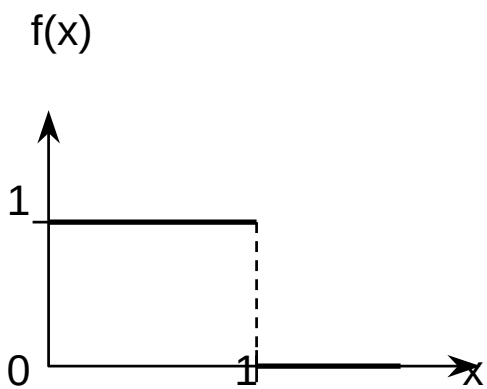
1. Since $F(\infty) = 1$, $\int_{-\infty}^{\infty} f(x)\,dx = 1$

2. Since F(x) is nondecreasing, $f(x) \geq 0$

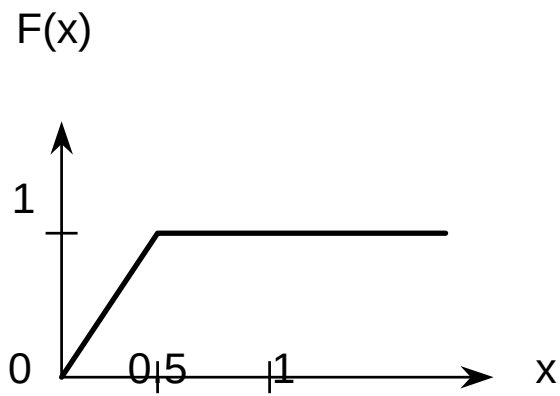
- No bounds on the value of f(x) at a point x.

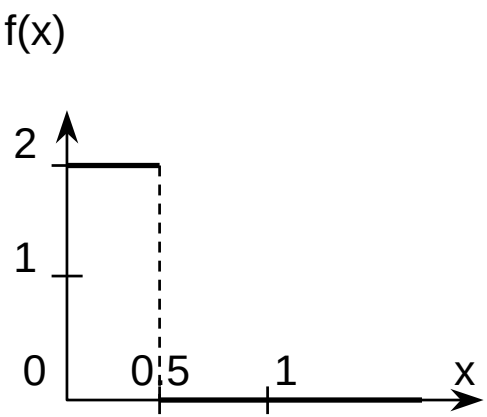
Uniform Dist.[0,1]





Uniform Dist.[0,0.5]





Calculating Probabilities

- The calculation of the probability of any event is simply a series of additions and subtractions of cumulative distribution.
- r.v. X $\{a < x \leq b\}$

$$P[x \leq b] = P[x \leq a] + P[a < x \leq b] = F(b) = \int_{-\infty}^{b} f(x)dx$$

$$P[x \leq b] - P[x \leq a] = P[a < x \leq b]$$

$$P[a < x \leq b] = F(b) - F(a) = \int_{a}^{b} f(x)dx$$

$$\text{pdf} \xrightarrow{\text{integration}} CDF \quad : \text{continuous case}$$

$$\text{pmf} \xrightarrow{\text{summation}} CDF \quad : \text{discrete case}$$

Calculating Expectation (평균, 기대치)

- Def. 19

  The expected value E[X] of a r.v. X is the weighted sum(integral) of all possible values of the r.v.

$$E[X] = \sum_{-\infty}^{\infty} x \cdot P_x(x) \qquad E[X] = \int_{-\infty}^{\infty} x \cdot f_x(x)\,dx$$

- 예) 주사위를 던져 얻는 수의 기대치

$$E[x] = 1\times\frac{1}{6} + 2\times\frac{1}{6} + 3\times\frac{1}{6} + 4\times\frac{1}{6} + 5\times\frac{1}{6} + 6\times\frac{1}{6} = \frac{21}{6} = 3.5$$

◆ 예) expected value of a r.v. with modified geometric distribution

p : 실패 확률, (1-p) : 성공확률

$$E[K] = \sum_{k=0}^{\infty} k(1-p)\cdot p^k$$

$$= \sum_{k=1}^{\infty} k(1-p)\cdot p\cdot p^{k-1} = (1-p)p\sum_{k=1}^{\infty} k\cdot p^{k-1}$$

$$= (1-p)p\sum_{k=1}^{\infty}\frac{d}{dp}p^k = (1-p)p\frac{d}{dp}\sum_{k=1}^{\infty}p^k$$

$$= (1-p)p\frac{d}{dp}\left(\frac{p}{1-p}\right) = (1-p)p\frac{1}{(1-p)^2} = \frac{p}{1-p}$$

- 예) expected value of an exponentially distributed r.v.

$$E[T] = \int_0^\infty t \cdot \lambda e^{-\lambda t} dt$$

$$= \int_0^\infty \lambda \cdot \left(t e^{-\lambda t} \right) dt = -\lambda \int_0^\infty \left(\frac{d}{d\lambda} e^{-\lambda t} \right) dt$$

$$= -\lambda \frac{d}{d\lambda} \int_0^\infty e^{-\lambda t} dt = -\lambda \frac{d}{d\lambda} \left[\frac{1}{-\lambda} \cdot e^{-\lambda t} \right]_0^\infty$$

$$= -\lambda \frac{d}{d\lambda} \left(\frac{1}{\lambda} \right) = -\lambda \cdot \left(-\frac{1}{\lambda^2} \right) = \frac{1}{\lambda}$$

- The expected valued can be calculated on a function of a r.v. as well as a r. v.
  - E[x] first moment, expected value, mean
  - $E[x^2]$ second moment, expected value, mean
  - $E[x^3]$ third moment, expected value, mean
  - $E[x^3 - x]$ .....

Multiple Random Variables

- Def. 20

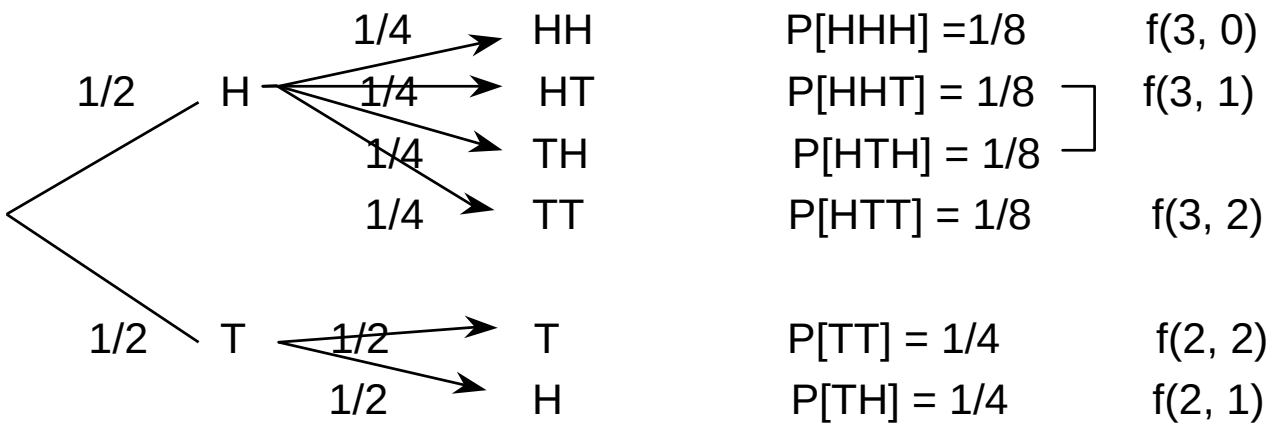
  The joint cumulative distribution function F(x, y) is a function defined on all values of r.v. X and Y.

  F(x,y) is equal to the probability measure such as

  $$F(x,y) = P[X \leq x \text{ and } Y \leq y]$$

- For the two r.v. case, the function is a surface that is nondecreasing in all the direction.

  - 예) 동전을 한 번 던져 T가 나오면 한번 더 던지고, H가 나오면 두 번 더 던질 경우, C를 동전 던지는 횟수, X를 C만큼 던지는 동안 나온 T의 횟수라고 하자.



$$F(c,t): F(2,0) = 0$$

$$F(2,1) = P[TH] = \frac{1}{4}$$

$$F(2,2) = P[TH] + P[TT] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

$$F(3,0) = P[HHH] = \frac{1}{8}$$

$$F(3,1) = P[HHH] + P[HHT] + P[HTH] + F(2,1) = \frac{5}{8}$$

$$F(3,2) = P[HTT] + P[HTH] + P[HHT] + P[HHH] + F(2,2) = 1$$

$$f(x,y) = \frac{\partial^2}{\partial x \partial y} F(x,y)$$

Marginal density function

- Def. 21

  The marginal pdf f(x) for a r.v. X is the integral(sum) over all values of r.v. Y of the joint pdf f(x,y)

$$f(x) = \int_{-\infty}^{\infty} f(x,y)dy$$

  - 예) 전장의 동전 던지기 예제

$$f(c) = f(c,0) + f(c,1) + f(c,2)$$

$$f(2) = f(2,0) + f(2,1) + f(2,2) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

$$f(3) = f(3,0) + f(3,1) + f(3,2) = \frac{1}{8} + \frac{1}{4} + \frac{1}{8} = \frac{1}{2}$$

- Marginal pmf

$$P_x(x) = \sum_{all\ y} P(x,y)$$

Conditional pdf

- Def. 22

  The conditional pdf f(x|A) of a r.v. X given an event A is the pdf f(x) restricted to the range of the r.v. within the event A, divided by the probability of the event A.

$$f(x|A) = \frac{f(x)}{P[A]}, \qquad x \in A$$

- Def. 22`

  Let X,Y be continuous r.v. with joint pdf f(x,y)

  The conditional pdf $f_{Y|X}$ is defined by

$$f_{Y|X}(y|x) = \frac{f(x,y)}{f_x(x)}, \qquad \text{if } 0 < f_x(x) < \infty$$

Conditional pmf

- Def. 23

Let X and Y be discrete r.v. having a joint pmf p(x,y).
The conditional pmf of Y given X is defined by:

$$P_{Y|X}(y|x) = P[Y = y | X = x] = \frac{P[Y = y \ \ and \ \ X = x]}{P[X = x]} = \frac{P(x,y)}{P_X(x)}$$

- Unconditioning

$$P_Y(y) = \sum_{all\ x} P(x,y) = \sum_{all\ x} P_{Y|X}(y|X) \cdot P_x(x)$$

$$f(y) = \int_0^\infty f(x,y)dx = \int_0^\infty f(y|x) \cdot f(x)dx$$

- Independence

Def. 24

Two r.v. are independent iff their joint pdf is equal to the product of their individual marginal probability density functions.

$f(x,y) = f(x) \cdot f(y)$

Reliability Modeling

1. Reliability 모델링 개요
2. Reliability Measures
3. Modeling Techniques
4. Evaluation Techniques

1. Reliability 모델링 개요

- Reliability(신뢰도)
  - 동작 시작 순간부터 어느 관찰 시점 T 까지 고장 없이 동작할 확률
- Dependability(신뢰성)
  - 시스템이 제공하는 서비스의 질, 정확성 및 연속성과 관련된 정성적 요소
  - Reliability, availability, safety, performability, maintainability를 종합적으로 지칭
- Reliability 분석의 필요성
  - 전자기기의 보편화
    - 사무자동화 시스템, 의료용 전자기기, 금융 전산망, 공장자동화 시스템
  - 시스템의 고장은 인명, 재산의 손실, 환경 파괴
- Dependability 향상 노력
  - Fault-tolerant design : redundant, parallel, distributed design
  - Reliability engineering

2. Reliability Measures

❑ 순간 계수

- Reliability, R(t)
  - A function of time defined as the conditional probability that the system will operate correctly during the interval [t_0 ,t], given that the system was operating correctly at t_o
  - 동작 시작 순간부터 시점 t 까지 고장 없이 동작할 확률
  - 고장에서 복구된 순간부터 그 후 시간 t 동안 계속 동작할 확률
- Instantaneous(point) availability, A(t)
  - 임의의 관찰시점 t 당시에 동작하고 있을 확률
- Safety, S(t)
  - Probability that a system will either operate correctly or will discontinue its function in a manner that does not disrupt the operation of other systems
  - 동작 개시부터 시점 t 까지 시스템이 불안전 상태에 처하지 않을 확률
    - (예) 하드디스크 시스템
    - 안전상태: 정상동작, 데이터 손실을 유발하지 않는 경미한 고장
    - 불안전상태: 저장된 테이터가 손상을 입는 고장이 발생

- Maintainability, M(t)
  - Probability that a failed system will be restored to an operational state within a period of time t
  - 고장이 발생시 정해진 복구 허용시간 내에 복구할 확률
  - 복구된 시스템이 재고장 시 복구 허용 시간 내에 복구를 마칠 확률

누적 계수

- Interval availability
  - 관찰시간 0~t 동안 성공적으로 동작한 기간의 비율
- Cumulative up-time(CUT)
  - 관찰시간 0~t 동안 성공적으로 동작한 기간의 합에 대한 기대치
- Cumulative down-time(CDT)
  - 관찰시간 0~t 동안 성공적으로 동작하지 않은 기간의 합에 대한 기대치
- Performability P(L, t)
  - Probability that the system performance will be at, or above some level L at time t

    → graceful degradation

❑ 시간 무관 계수

- Mean time to failure(MTTF)
  - 시스템이 고장나기까지 시간의 기대치
- Mean time to repair(MTTR)
  - 시스템이 고장난 후 수리되기까지 시간의 기대치
- Mean time between failure(MTBF)
  - 시스템이 고장들 간격 시간의 기대치
- Steady state availability
  - 시스템이 steady state 에 이른 후, 성공적으로 동작할 확률

Fault

A physical defect, imperfection that occurs within some hardware or software component of a system

예) A short between electrical conductors, open or break in conductors ,
physical flaws in semiconductor device, endless loop in a program

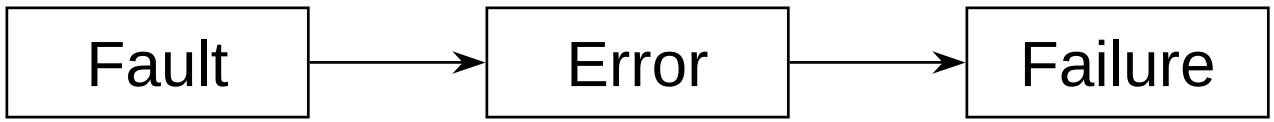
Error

Fault로 인해 accuracy 나 correctness의 훼손이 출현된 상태

예) Electrical device의 fault로 인해 line(circuit)의 logic이 0 이어야 하는데 1 로 되는 것

Failure

Error로 인해 system의 기능이 동작하지 않거나 기준 이하의 성능을 내는 상태

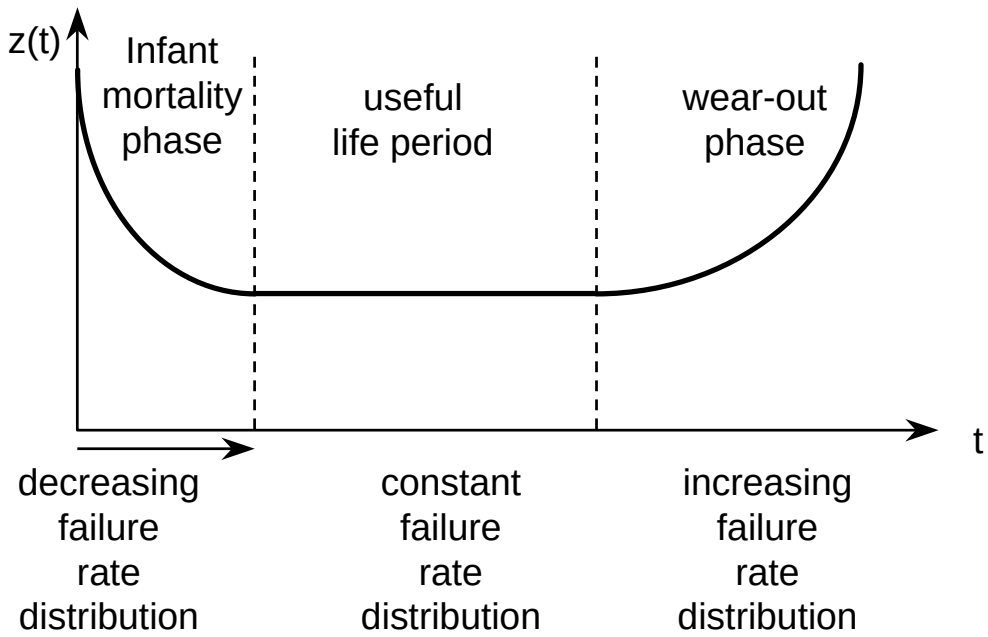


- Failure rate

  Expected number of failures of a system per a given time period

- Failure rate function, hazard function z(t)

$$z(t) = -\frac{1}{R(t)} \cdot \frac{dR(t)}{dt}$$



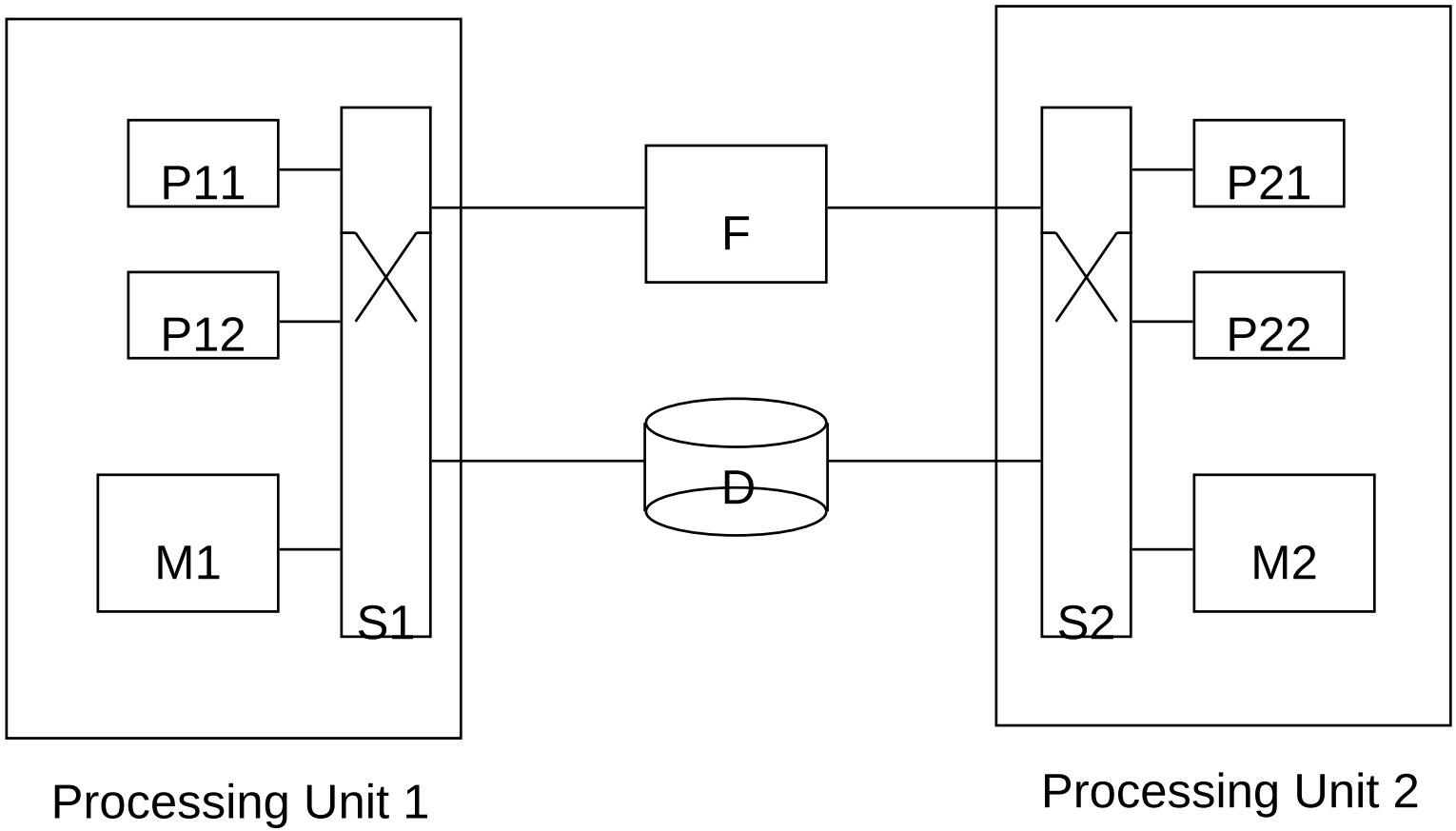
3. Modeling Techniques

- Dependability 모델링 기법
  - 상태 기반 모델(state-space model)
    - 시스템이 발생할 수 있는 event에 따르는 상태 변화를 표현
    - Markov model
    - Stochastic Petri-Nets
  - 비상태 기반 모델(non state-space model)
    - Reliability Block Diagram(RBD)
    - Reliability Tree
    - Reliability Graph

Reliability Block Diagram

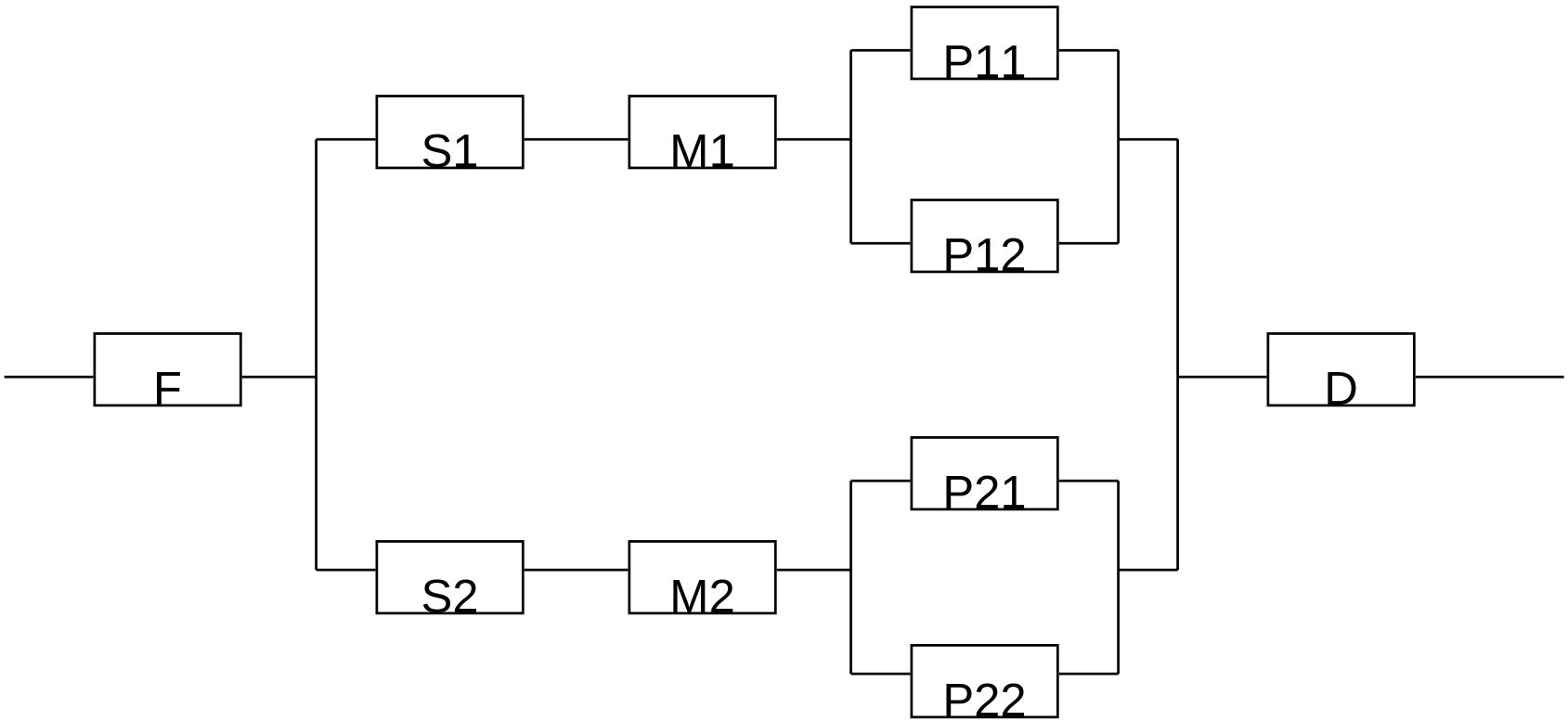
- RBD : 블럭과 블럭들을 연결하는 선분들로 이루어지는 그래프 구조
  - 블럭 : 시스템 컴포넌트
  - 선분 : 시스템 컴포넌트들 간의 연관성
- RBD는 시스템의 실제 하드웨어 구조를 표현하는 것이 아니고 시스템 동작면에서 각 컴포넌트들 간의 관계를 표현
- Structure Based Classification
  - Series Structure
    - 어느 하나의 컴포넌트가 고장이 나면 시스템 고장
    - MTTF(system) < MTTF(any components)
  - Parallel Structure
    - 적어도 하나의 컴포넌트라도 작동하면 시스템 작동
    - MTTF(system) > MTTF(all components)
  - kOFn Structure
    - n 개중 적어도 k 컴포넌트가 작동해야 시스템 작동
    - Generalization of the two structures described above. (1OFn, nOFn)
  - Series-Parallel Structure
    - Combination of the two above yields a series-parallel system

Fault-Tolerant DB 시스템



S: Switch F: Front-End Processor M: Memory P: Processor D: Database

DB 시스템의 RBD 모델



DB 시스템의 Reliability

$R_F(t)$: reliability of the front-end

$R_D(t)$: reliability of the database

$R_{Si}(t)$: reliability of the switches i

$R_{Mi}(t)$: reliability of the memory unit i

$R_{Pi1}(t)$, $R_{PI2}(t)$: reliability of the processor P_{i1}, P_{i2} (i =1,2)

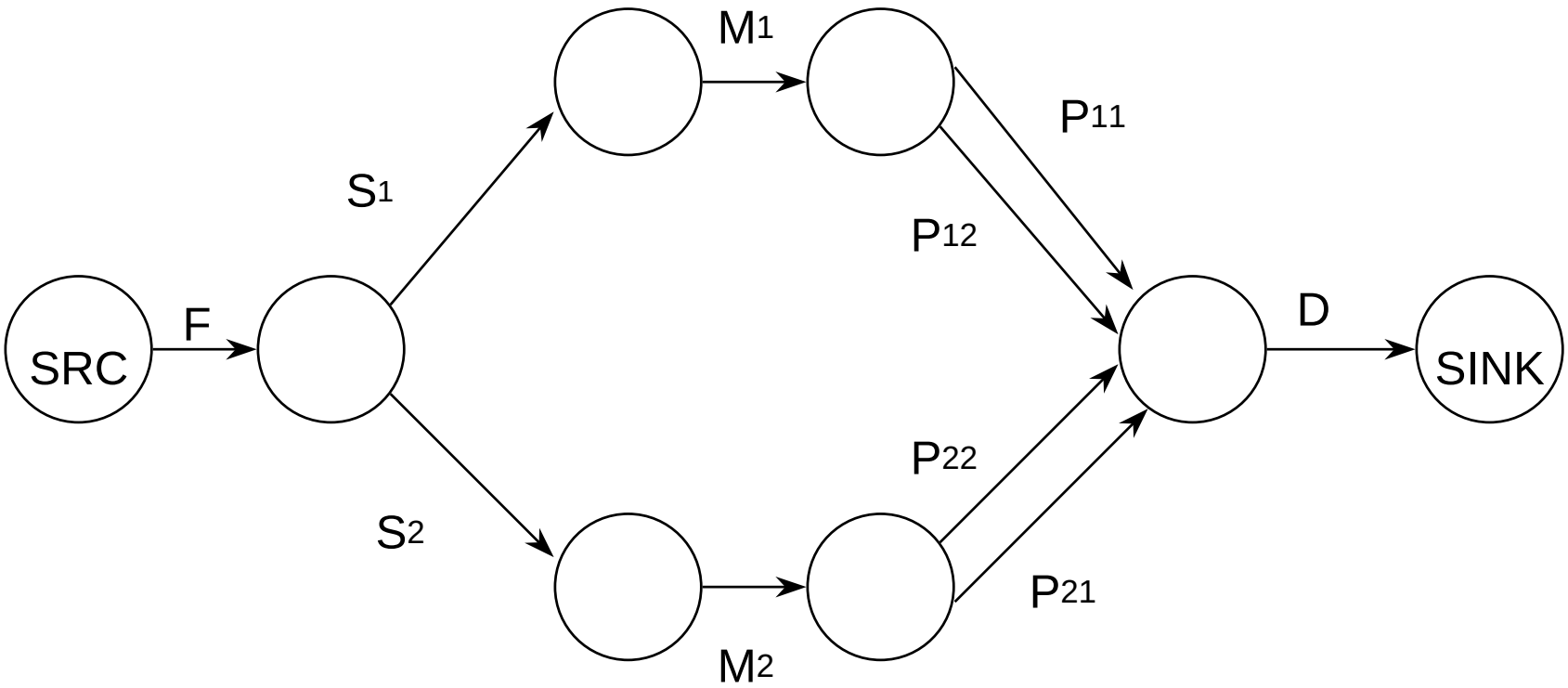
$$R_S(t) = R_F(t) * R_D(t) * \left(1 - \prod_{i=1}^{2} \{1 - R_{Si}(t) * R_{Mi}(t) * [1-(1- R_{Pi1}(t))*(1- R_{PI2}(t))] \} \right)$$

$$MTTF_S = \int_0^{\infty} R_S(t)dt$$

Reliability Graph(RG)

- 시스템 컴포넌트들 간의 동작 관계를 표시한 방향성 그래프로서 RBD를 확장시킨 기법
- 그래프의 방향성 성분
  - 시스템의 컴포넌트를 표시
  - 컴포넌트 고장 시 선분 단절
- 시스템의 Reliability
  - 시작 노드로부터 끝 노드까지 연결된 경로 존재 시 이상 없이 동작

DB 시스템의 RG 모델



RG 모델의 Input, Output

- Input Attached to Components
  - Probability of failure of a component
  - Distribution of time to failure of a component
  - Failure rate function of a component

- Output Measures Computed
  - Mean time to failure of the system
  - System reliability(or unreliability) at time ***t***
  - Steady-state or instantaneous availability of the system

RG 모델의 장점

- RGs model operational dependency of a system on its components.
- RGs are easy to understand.
- Given a description of operational dependency of a system on its components, the corresponding RG is easy to generate.
- It is easy to do a hierarchical modeling.
- Arbitrary time to failure distributions of system components are allowed.

RG 모델의 단점

- Stochastic dependency between different faults can not be modeled.
- RGs can not model dynamic system structure.
- The computational complexity of solving an arbitrary non-series-parallel RG could be exponential in the number of components.
- Details of the fault recovery behavior are not captured by RGs
- Only one type of fault for each component is modeled.
- RGs can not easily model different system failure modes.
- RGs may not be used for dependability analysis of repairable systems.
- Common-mode failures are not easily modeled by RGs.
- RGs have been traditionally used for block structured systems where simple failures are assumed.

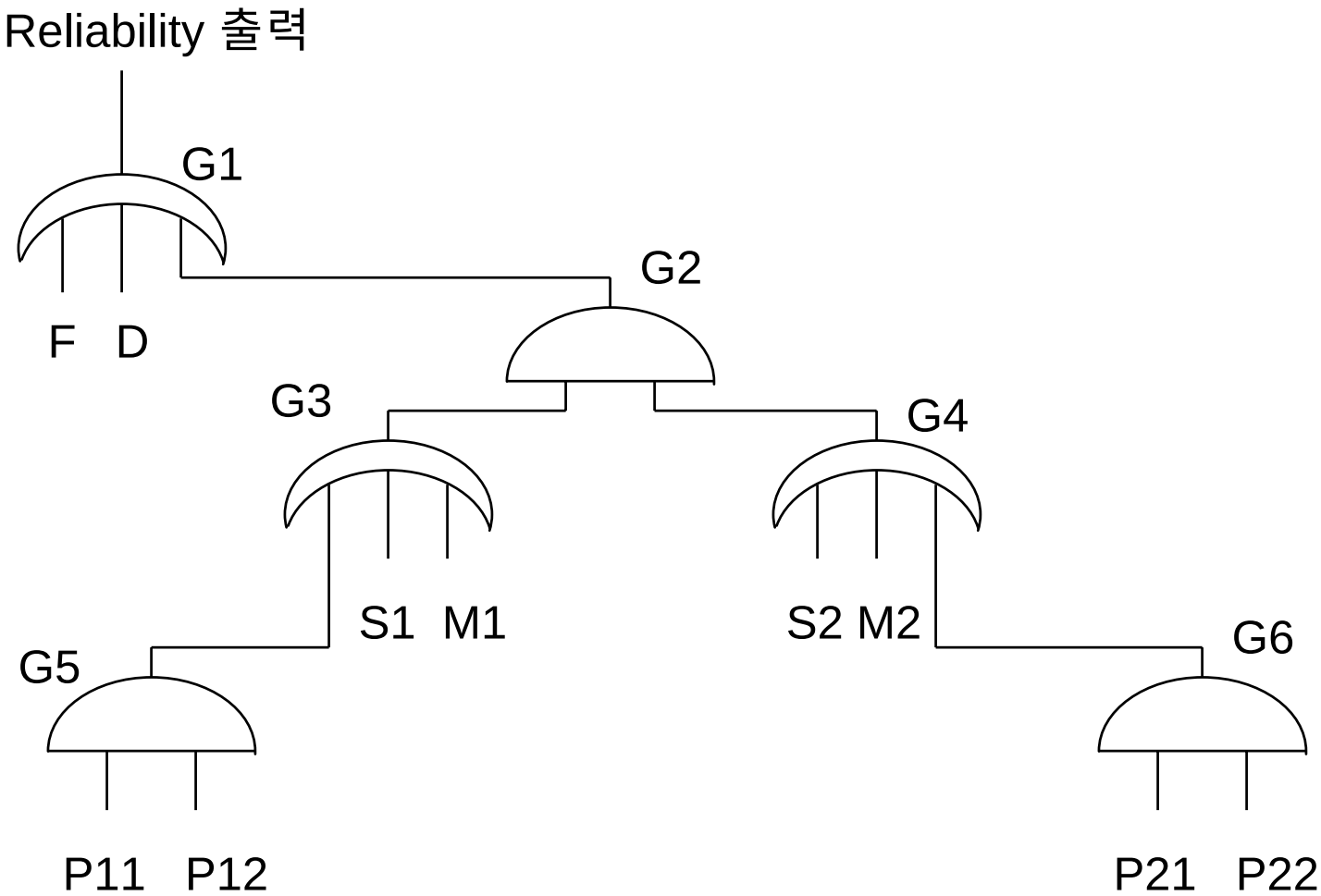
Fault Tree(FT)

- Fault tree 란 AND, OR, NOT 등의 논리 게이트들에 시스템 컴포넌트들이 입력으로서 연결된 트리
  - 논리게이트의 입,출력
    - 해당 컴포넌트의 고장 상태 표현(TRUE, FALSE)
  - 컴포넌트가 고장이 나면 TRUE 값이, 정상 동작하면 FALSE 값이 그 컴포넌트 노드에 연결된 선분에 할당되어 게이트에 입력
  - 루트 노드에서의 논리 값
    - 그 순간에 시스템이 정상 동작하는지 고장인지를 표시
  - FT는 시스템의 실제 구조를 나타내는 것이 아니고 시스템 고장을 유발하는 관계를 표현

FT의 Gate 정의

- AND : 모든 입력이 fail(TRUE) 이면 출력이 TRUE, 그 외는 FALSE
- OR : 하나의 입력이라도 fail(TRUE) 이면 TRUE, 그 외는 FALSE
- NOT : 입력이 fail 이면 출력이 FALSE, 그 외는 TRUE
- kOFn : n개의 입력 중 적어도 k개가 fail이면 TRUE, 그 외는 FALSE
- PAND(priority AND) : 모든 입력이 정해진 순서대로 fail되면 TRUE
- Inhibit : 2개의 입력만 허용
  - 2개의 입력: inhibit event(enabling condition) 와 component로부터의 입력
  - inhibit event가 TRUE 이면 출력은 FALSE
  - inhibit event FALSE 이고 component가 fail이면 출력은 TRUE
- XOR : 오직 한 입력이 fail되면 TRUE, 그 외는 FALSE

DB 시스템의 FT 모델



FT 모델의 Input, Output

- Input Attached to Basic Events
  - Probability of failure of a component.
  - Failure rate of a component.
  - Distribution of time to failure of a component.
  - Combination of the above.
  - Instantaneous unavailability of a component.
  - Steady state unavailability of a component.
  - Time that a certain component is operational.

- Output Measures Computed
  - Mean time to failure of the system.
  - System failure probability by time t,
  - Instantaneous unavailability at time t.
  - Steady-state unavailability of the system.

FT 모델의 장점

- Fault trees model the conditions leading to system failure.
- Arbitrary failure distributions are allowed.
- Very large systems can be concisely described.
- Use of kOFn gate provides a useful abbreviation.
- A fault tree model is easy to understand.
- Recent research efforts have made it possible to model sequence dependent failures using FTs.
- A fault tree model provides useful information about events leading to system failure.
- Fault-tree models are useful in hierarchical modeling of large and complex systems.

FT 모델의 단점

- Stochastic dependence of faults can not be modeled.
- Fault recovery models may not be modeled.
- Complex system behavior can not be modeled: not for repairable systems with a shared repair facility
- Complex FT models may require solution time exponential in the size of the tree.
- Dynamic system structure can not be modeled.

Markov Model

- Dependability measures are expressed as weighted sums of state probabilities.
- State space S is partitioned into
  - S_U : Up states, operational states
  - S_D : Down states, failure states
- Instantaneous availability
  - $I(t) = \begin{cases} 1 & \text{if } X(t) \in S_u \\ 0 & \text{otherwise} \end{cases}$ *where* $X(t)$: State of the MC at time t
  - $P_i(t)$: Probability of the system being in state i at time t

$$A(t) = P[I(t) = 1] = \sum_{i \in Su} Pi(t)$$

- Reliability

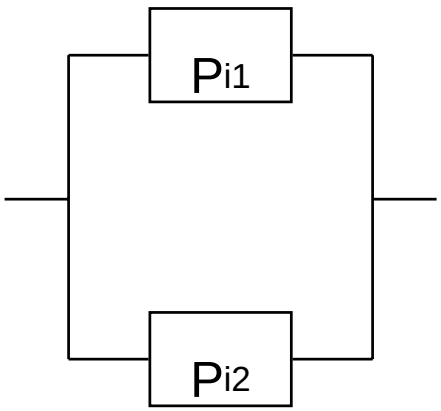
$$R(t) = P[I(x) = 1, \forall x \in [0, t]]$$

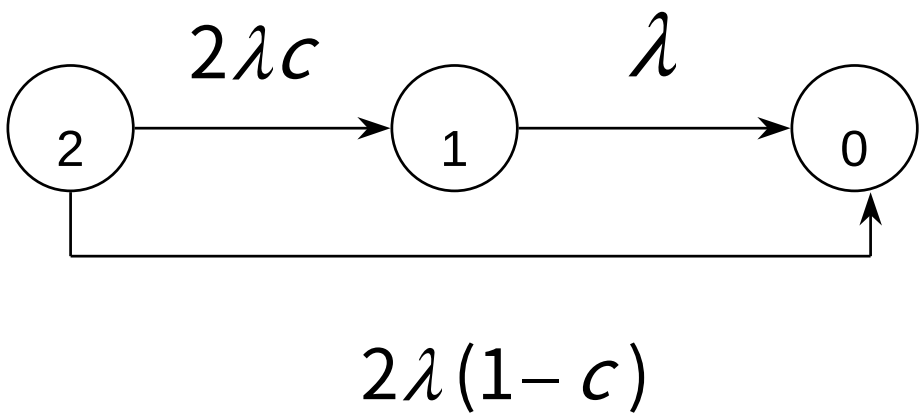
  - When S_D are absorbing states.

$$R(t) = \sum_{i \in Su} Pi(t)$$

CTMC 모델(1)

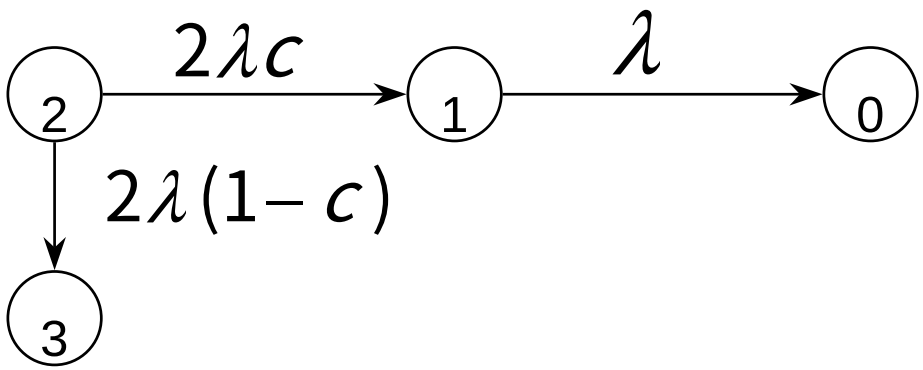
- Two processor subsystem with imperfect coverage.



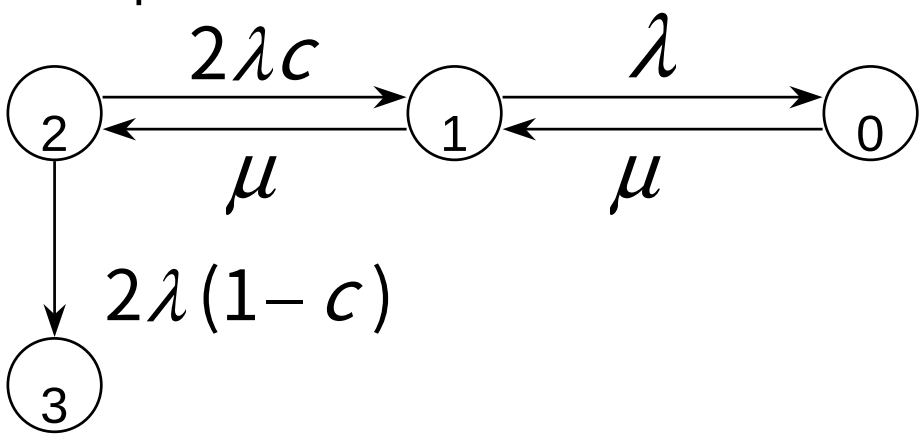


CTMC 모델(2)

- Two processor subsystem with multiple absorbing states.

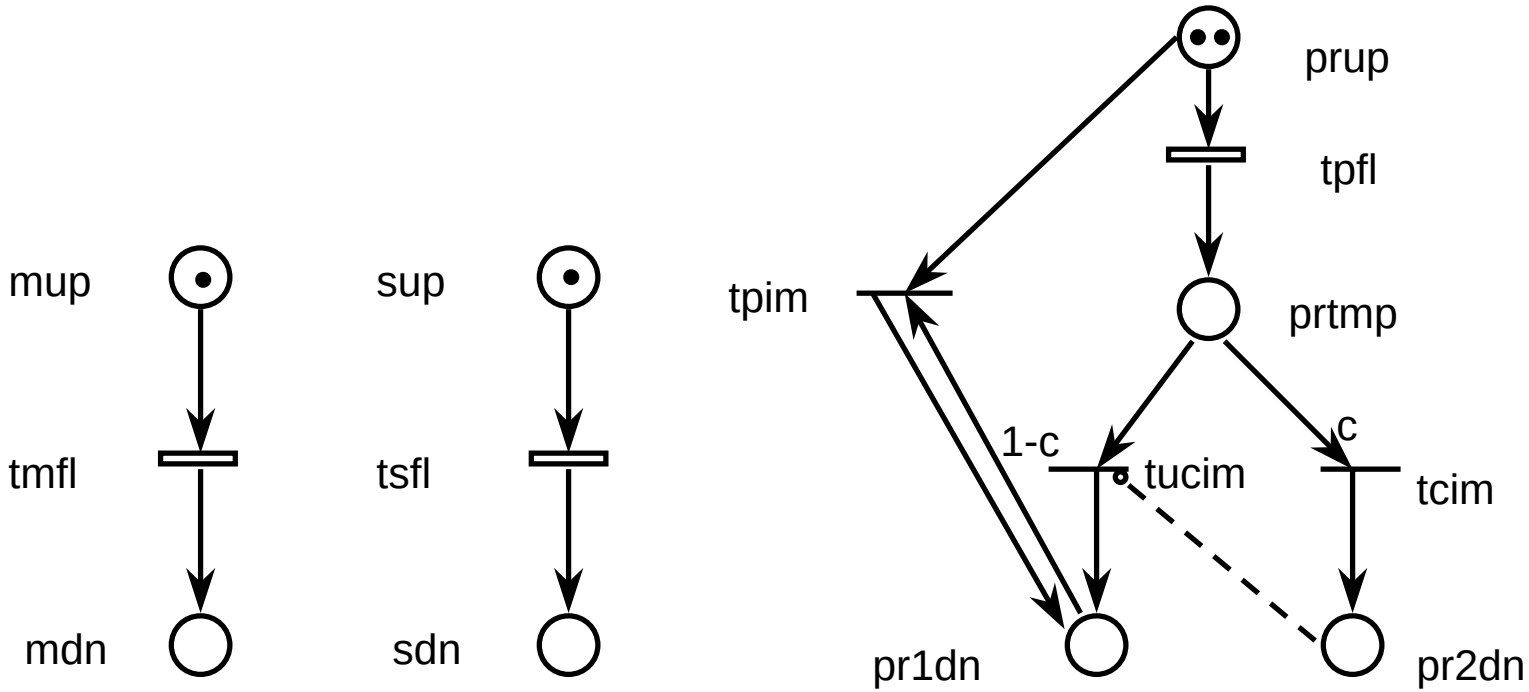


- Failure-repair model



Petri Nets Reliability Model

Marking	Reward Rate Function
(mup == 0) || (sup == 0) || (prup == 0)	0
otherwise	1



4. Evaluation Techniques

Mean Time To Failure(MTTF)

- Expected time that a system will operate before the first failure occurs.
  Expected time until the next failure after a repair.
- N identical systems, each system operate for t_i before failure.

$$MTTF = \frac{t_1 + t_2 + t_3 + \cdots + t_N}{N}$$

$$MTTF = E[T] = \int_0^\infty t \cdot f_T(t)dt = \int_0^\infty R(t)dt$$

- $F(t) = P[T \le t]$
  $R(t) = P[T > t] \qquad \therefore R(t) = 1 - F(t)$

예) exponential failure law

$$F(t) = 1 - e^{-\lambda t}, \qquad R(t) = e^{-\lambda t}$$

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda}$$

Mean Time To Repair(MTTR)

- Average time to repair a system
- N faults, t_i : repair time of i-th fault

$$MTTR = \frac{t_1 + t_2 + t_3 + \cdots + t_N}{N}$$

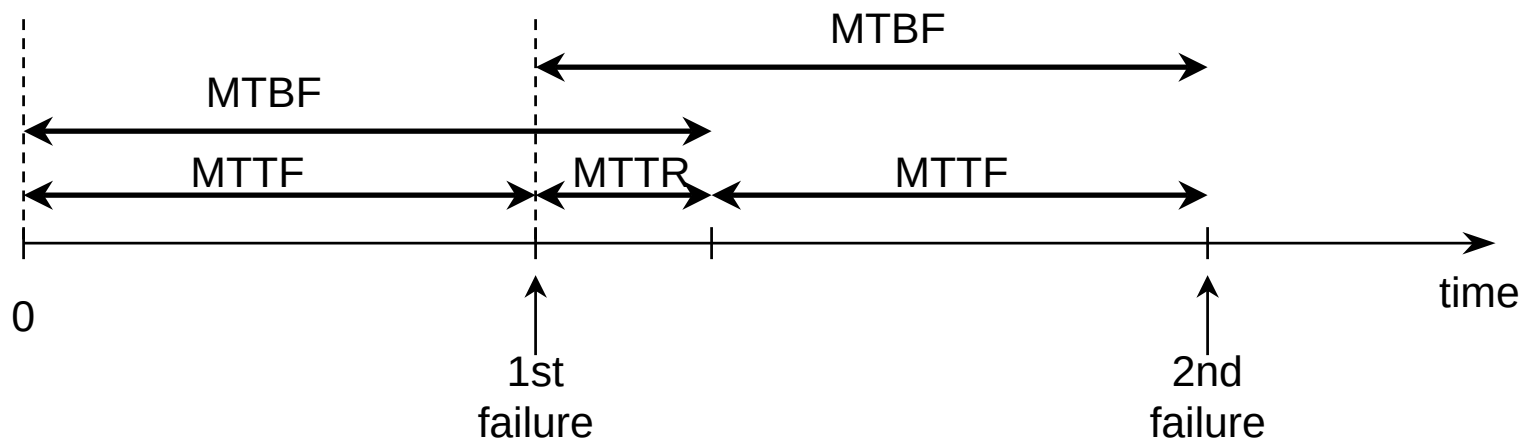
- Exponentially distributed repair time

Repair rate μ : average number of repairs that occur per time period

$$\text{MTTR} = \frac{1}{\mu}$$

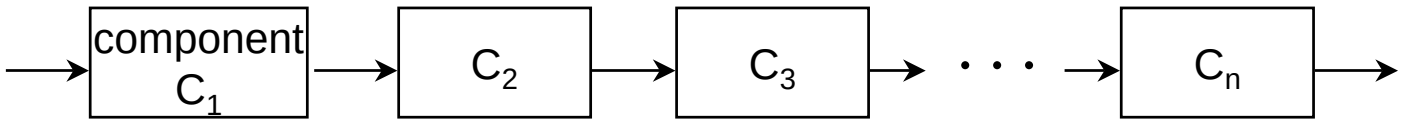
Mean Time Between Failure(MTBF)

- Average time between failures of a system
- MTBF = MTTF + MTTR



Reliability of Series Systems

- Series system contains no redundancy
- A reliability block diagram



$R_i(t)$: Reliability of component C_i at t

$W_i(t)$: An event that component C_i is working at t

$$R_{series}(t) = P\left[W_1(t) \cap W_2(t) \cap W_3(t) \cap \cdots \cap W_n(t)\right]$$

assuming C_i are <u>independent</u>

$$R_{series}(t) = R_1(t) \times R_2(t) \times R_3(t) \times \cdots \times R_n(t)$$

$$= \prod_{i=1}^{n} R_i(t)$$

- 예) $R_1(t) = R_2(t) = R_3(t) = 0.9$

  $R_{series}(t) = 0.9 \times 0.9 \times 0.9 = 0.729$

- 예) A series system of components that satisfy the exponential failure law

$$R_{series}(t) = R_1(t) \times R_2(t) \times \cdots \times R_n(t) = e^{-\lambda_1 t} \cdot e^{-\lambda_2 t} \cdots e^{-\lambda_n t}$$

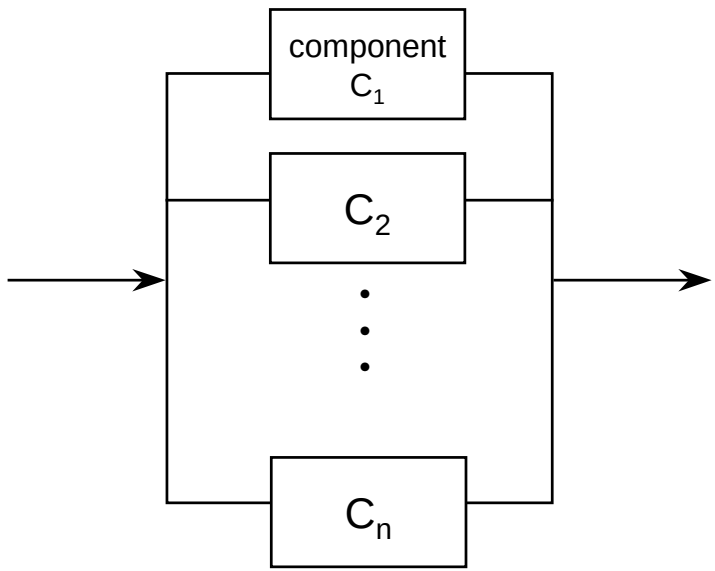
$$= e^{-(\lambda_1 + \lambda_2 + \cdots + \lambda_n)t} = e^{-\sum_{i=1}^{n} \lambda_i t} = e^{-\lambda_{sy} \cdot t}$$

$$where \ \lambda_{sy} = \sum_{i=1}^{n} \lambda_i$$

Reliability of Parallel Systems

- Only one of n identical components is required for the system to function

- Reliability Block Diagram



$R_i(t)$: Reliability of component C_i at t

$O_i(t)$: Event that the component C_i has failed at t

$U_{para}(t)$: Unreliability of the parallel system at t

$U_i(t)$: Unreliability of component the C_i at t

$$U_{para}(t) = P\left[O_1(t) \cap O_2(t) \cap O_3(t) \cap \cdots \cap O_n(t)\right]$$

assuming independence

$$U_{para}(t) = U_1(t) \times U_2(t) \times \cdots \times U_n(t) = \prod_{i=1}^{n} U_i(t)$$

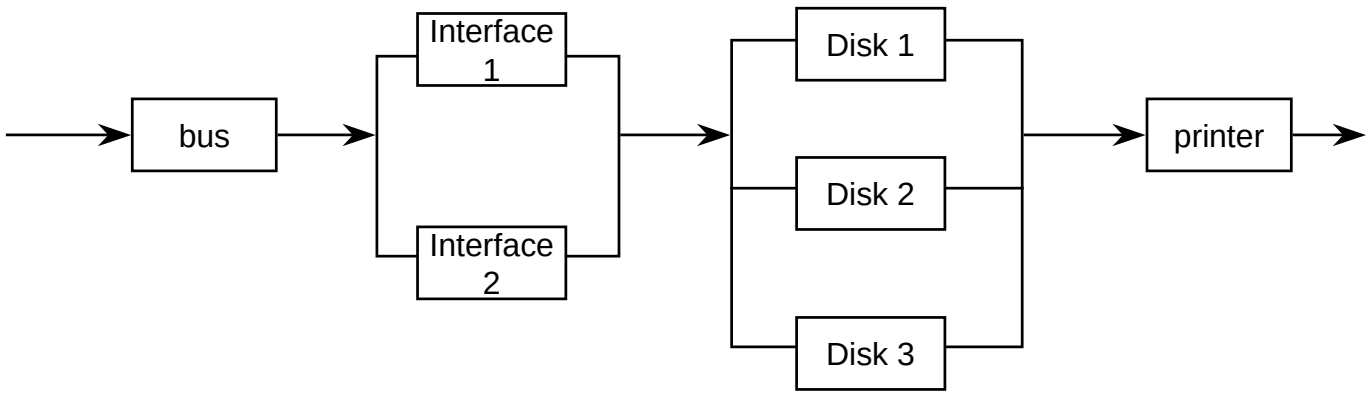
$$R_{para}(t) = 1 - U_{para}(t) = 1 - \prod_{i=1}^{n} U_i(t) = 1 - \prod_{i=1}^{n} \left(1 - R_i(t)\right)$$

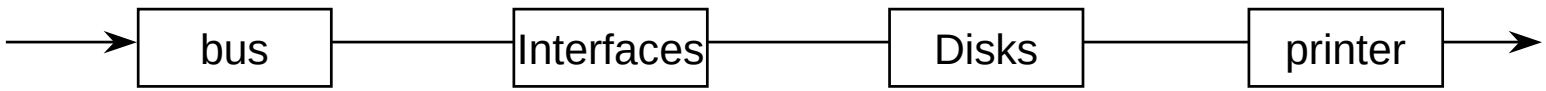
예) $R_1(t) = R_2(t) = R_3(t) = 0.9$

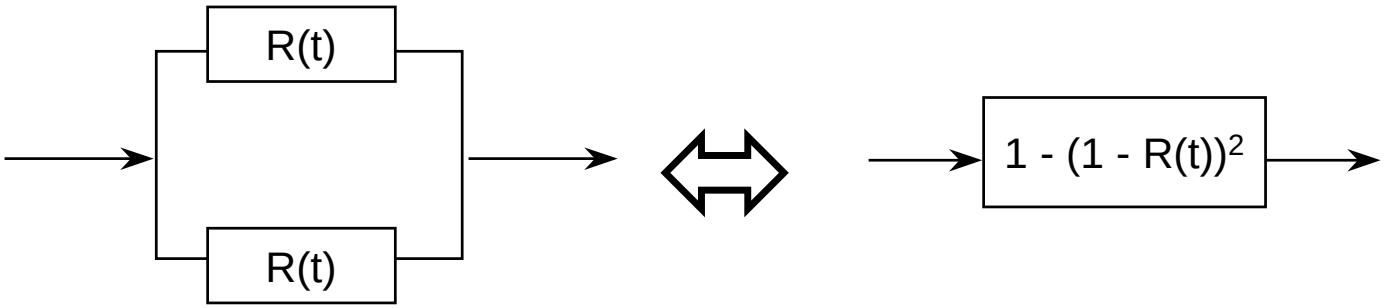
$R_{para}(t) = 1 - (1 - 0.9) \times (1 - 0.9) \times (1 - 0.9) = 0.999$

Reliability of Series/Parallel Systems

- A system that contains both series and parallel structure
- Reduce the reliability block diagram to a single series diagram by replacing parallel portions with an equivalent single element



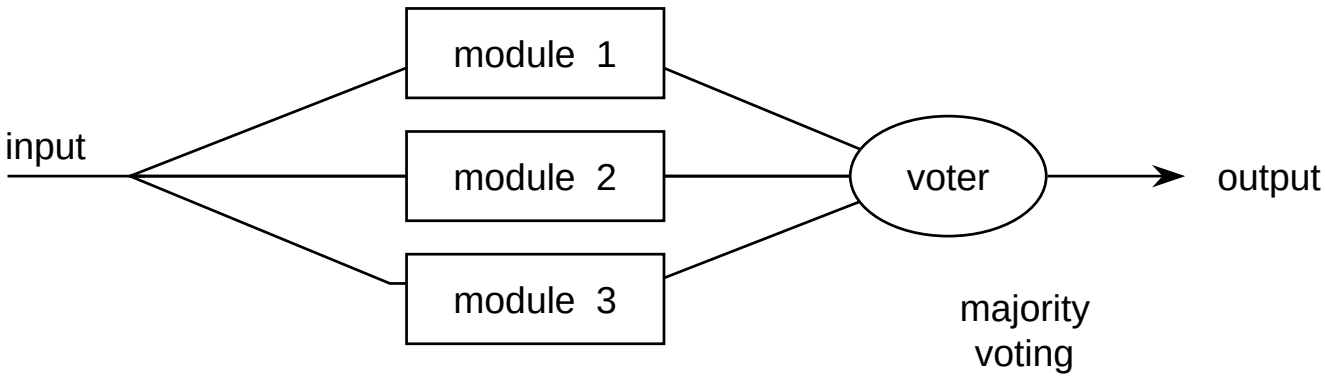




$$R_{system}(t) = R_{bus}(t) \times \left[1 - \left\{1 - R_{IF}(t)\right\}^2\right] \times \left[1 - \left\{1 - R_{Disk}(t)\right\}^3\right] \times R_{\text{printer}}(t)$$

M-of-N Systems

- M of the total N identical modules are required to function
- Generalization of the parallel systems
- 2-of-3 system : TMR(Triple Modular Redundancy) System

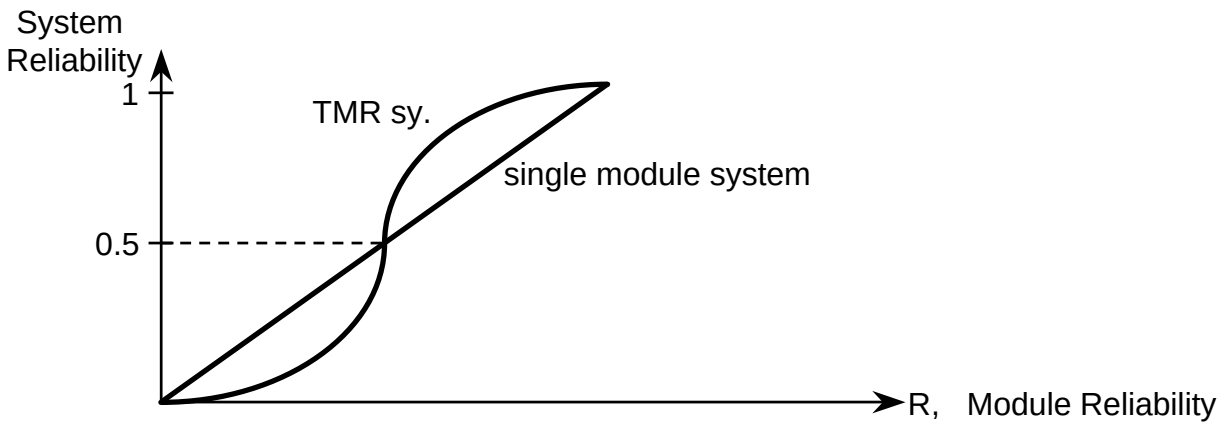


- Let $R_{TMR}(t)$: Reliability of TMR system
  $R_i(t)$: Reliability of module i

$$R_{TMR}(t) = R_1(t)R_2(t)R_3(t) + R_1(t)R_2(t)\{1 - R_3(t)\} + R_1(t)\{1 - R_2(t)\}R_3(t) + \{1 - R_1(t)\}R_2(t)R_3(t)$$

if $R_1(t) = R_2(t) = R_3(t) = R$

$$R_{TMR}(t) = R^3 + 3R^2\{1 - R\} = 3R^2 - 2R^3$$



crosspoint

$R_{TMR} = 3R^2 - 2R^3 = R$

$2R^2 - 3R + 1 = 0$

$R = 0.5 \text{ or } 1$

- 개별 component(module)의 reliability 가 0.5인 TMR시스템은 fault tolerance는 있으나 reliability는 single component system 보다 나을 것이 없다.
- TMR system이 single module system 보다 reliability가 높으려면 개별 component의 신뢰도가 0.5보다 커야 한다.

$\mathrm{M-of-N\ system\ reliability}$

$$R_{M-of-N}(t) = \sum_{i=0}^{N-M} \binom{N}{i} R^{N-i}(t)\{1-R(t)\}^i$$

$$\binom{N}{i} = \frac{N!}{(N-i)!\ i!}$$

Markov Models

1. Introduction
2. State Classification
3. DTMC
4. CTMC

1. Introduction

- Modeling a system's behavior of a discrete state space system

  : Characterize the dependency of a possible state on the sequence of states S through which the system has previously passed.

  $S = S_1, S_2, S_3, S_4, \ldots, S_n, S_{n+1}, \ldots$

  $S = \{m_1, m_2, m_3, m_4, \ldots, m_k\}$

  $P[S_{n+1} = m_i \mid S_n = m_j , S_{n-1} = m_1 , S_{n-2} = m_k , \ldots , S_1 = m_5]$

  - To determine the probability of a particular state being the next state.
  - To determine the state which the system visits most frequently.
  - To determine the state in some time *t* after the system begins from a initial state S_1

- May use simulation for these purpose. But the simulation becomes too complex to count all the different history.

- Stochastic process
  - ① Discrete time processes : the processes are embedded at separate points in time.
  - ② Continuous time processes : the processes that change states at an arbitrary point in time.

- Different levels of complexity of stochastic processes are determined by the dependencies between state changes.
  - The next state depends on its entire past history of state changes.
  - The next state depends on the last state change.
  - The next state depends on what the last state was.

- Types of discrete state space stochastic processes that depend only on the history from the last state change

1. Semi - Markov process
   - The next state may be any one in the state space.
   - The next state probabilities depend on the current state and are arbitrary.
   - Arbitrary distribution of time between state changes.

2. Random walk
   - The next state may be any one in the state space.
   - The next state probabilities depend on the distance from the current position.

$$P\left[S_{n+1} = m_k \,|\, S_n = m_j\right] = P_{k-j}$$

   - Arbitrary distribution of time between state changes.

3. Markov chains

- The next state may be any one in the state space.
- The next state probabilities depend on the current state and are arbitrary.
- Exponential distribution of time between state changes

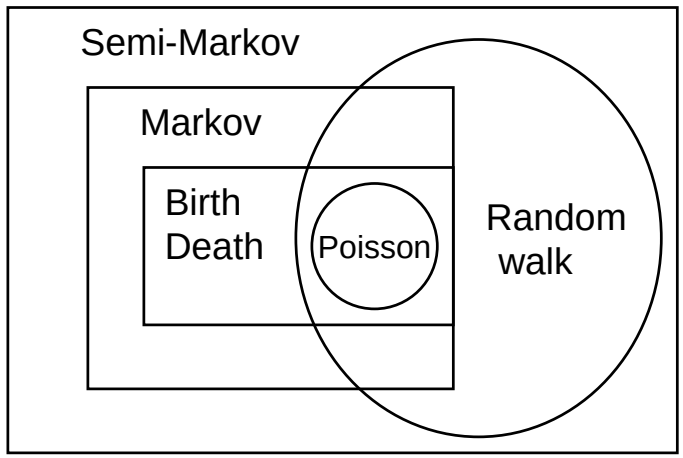
  for the continuous time Markov chains(CTMC),

  Geometric distribution

  for the discrete time Markov chains(DTMC).

4. Birth - Death process

- The next states are the nearest-neighbor state.
- The next state probabilities are zero if the next state is not a neighbor state of the current state.
- Exponential/geometric distribution of time between state changes.



Different Types of Stochastic Processes

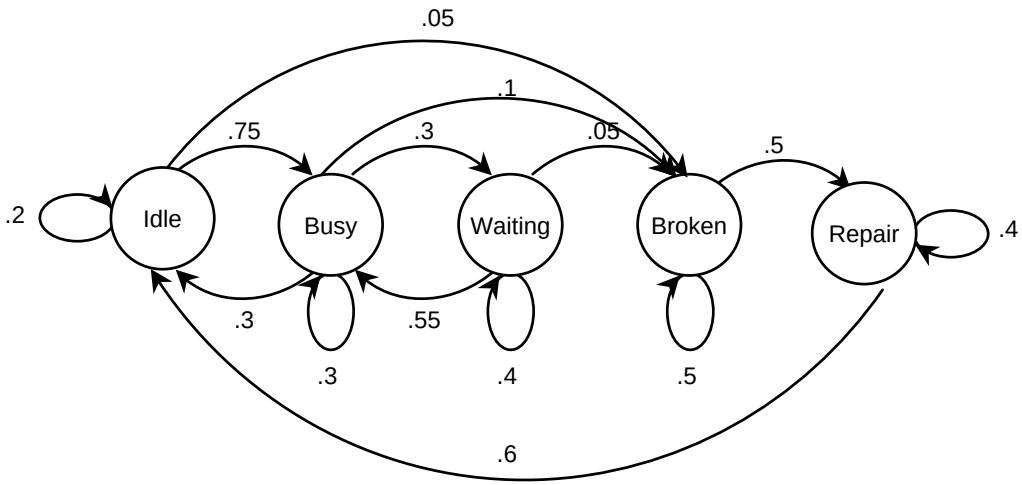
- Example of a Markov chain : a model of a processor
  - State space S = {idle, busy, waiting, broken, in repair}
  - Possible state changes

Current State	Possible Next State	Possible Next State	Possible Next State
Idle	Busy	Broken	-
Busy	Idle	Waiting	Broken
Waiting	Busy	Broken	-
Broken	Repair	-	-
Repair	Idle	-	-

- Transition probability matrix is a stochastic matrix

States	Idle	Busy	Waiting	Broken	Repair
Idle	0.20	0.75	0.00	0.05	0.00
Busy	0.30	0.30	0.30	0.10	0.00
Waiting	0.00	0.55	0.40	0.05	0.00
Broken	0.00	0.00	0.00	0.50	0.50
Repair	0.60	0.00	0.00	0.00	0.40

- Graphical representation of a Markov chain



- Example of a Birth Death process
  - State space S = {0, 1, 2, 3, ... }
  - Possible state change

    for state 0, next state is 0 or 1

    for all other state i, next state is i-1, i, i+1

  - Transition probability matrix

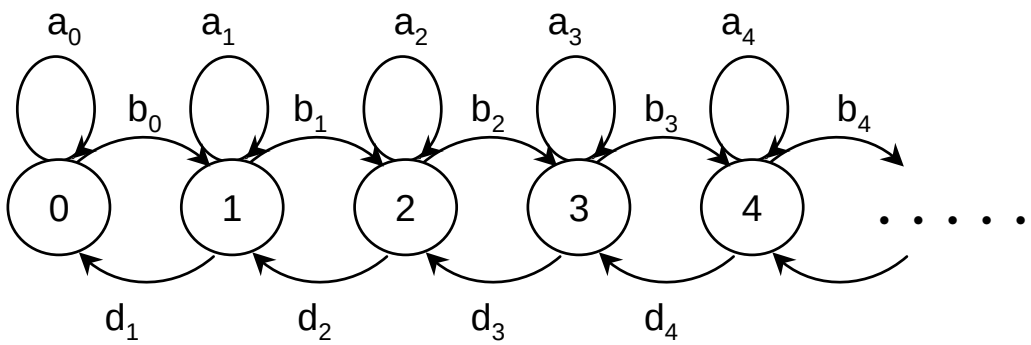
Stochastic matrix

$a_0 + b_0 = 1$

$a_i + b_i + d_i = 1, \quad for \ \ i \geq 1$

$$P = \begin{bmatrix} a_0 & b_0 & 0 & 0 & \varnothing \\ d_1 & a_1 & b_1 & 0 & \\ 0 & d_2 & a_2 & b_2 & \\ 0 & 0 & d_3 & a_3 & b_3 \\ \varnothing & & & & \ddots \end{bmatrix}$$

- State diagram



2. State Classification

Def. 6.1

A state is recurrent if the system is in that state and will return to that state through a series of transitions with probability 1.

Def. 6. 2

A state is transient if it is not recurrent. (The probability of returning to that state is less than 1)

- Transient state도 recur할 수 있으나 반드시 recur된다고 보장못함.
- Recurrent state는 recur하는데 걸리는 시간에 따라 두 가지로 분류.
  - Recurrent nonnull
  - Recurrent null

Def. 6.3

A recurrent state is recurrent nonnull if the mean time to return to the state is finite.

Def. 6. 4

A recurrent state is recurrent null if the mean time to return to the state is infinite.

- The number of states in a system with recurrent null states is infinite.
- In a finite state space Markov chain, there are no recurrent null states and not all states can be transient.
- Recurrent state는 recur하는 형태에 따라 아래와 같이 분류.
  - Aperiodic state
  - Periodic state

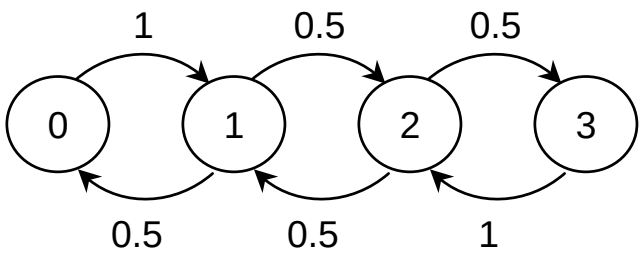
Def. 6.5

A recurrent state is aperiodic if, for some number k, there is a way to return to the state in k, k+1, k+2, ..., ∞ transitions.

Def. 6.6

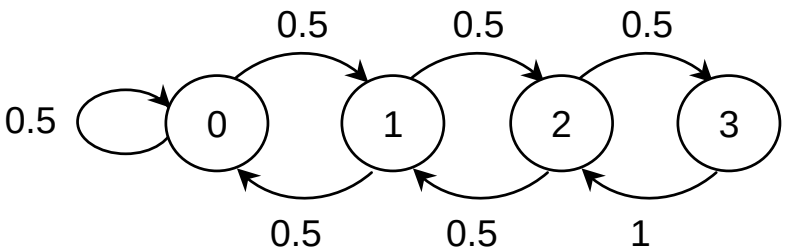
A recurrent state is periodic if it is not aperiodic.

- Example of a Markov chain with periodic states



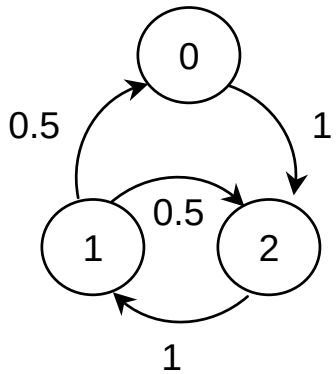
All states are recurrent, nonnull and periodic.

- Example of a Markov chain with aperiodic states



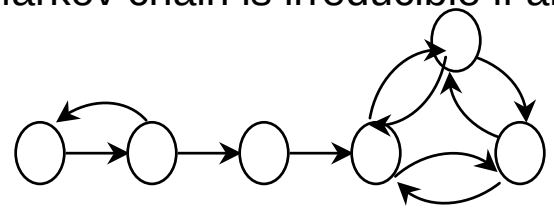
All states are recurrent, nonnull and aperiodic.

- Not having a self-loops does not mean that a state is periodic.

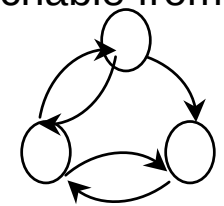


Def. 6.7

A Markov chain is irreducible if all states are reachable from all other states.



reducible MC



irreducible MC

Theorem 6.1

All states of an irreducible, finite state MC are of the same type.

예) One state of an irreducible MC is aperiodic → All states are aperiodic.

예) One state of an irreducible MC is periodic.

→ All states are periodic with the same period.

예) One state of an irreducible MC is recurrent.

→ All states are recurrent.

* Irreducible MC 가 infinite state space를 가질 때 transient states를 가질 수 있다.

Def. 6.8

A Markov chain is called

transient if all its states are transient.

recurrent nonnull if all its states are recurrent nonnull.

recurrent null if all its states are recurrent null.

periodic(aperiodic) if all its states are periodic(aperiodic).

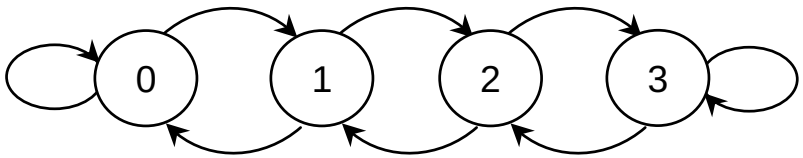
Def. 6.9

If a Markov chain is irreducible, recurrent nonnull and aperiodic, it is called ***ergodic***.

Theorem 6.2

If a Markov chain is ergodic, there exists a unique limiting distribution for the probability of being in a state k denoted by π_k independent of the initial state. (steady state probability equilibrium probability)

예) ergodic MC



$\pi = (\pi_0, \pi_1, \pi_2, \pi_3)$: *steady state probability vector*

$$\pi_i = \lim_{t \to \infty} P[X(t) = i]$$

3. DTMC

- Markov chain model of a system
  - Operation of a system → (possibly infinite) sequence of states
  - The way of predicting system operation.
    → not a specific sample path of the Markov process
    but the relative frequency of the different outcomes

- State probability vector P=(p_i)

  p_i : (mean) probability of finding the system in state i

- 예) Communication interface that is either busy or idle.
  - 어느 순간에 system이 busy일 때 P = (p_{busy} , p_{idle}) = (1,0)
  - 어느 순간에 system이 idle일 때 P = (0,1)
  - 여러 번 관찰한 결과, (평균적으로) 1/3 시간 동안은 busy, 2/3시간 동안은 idle이라면, P = (1/3 , 2/3)

- One-step transition probability matrix P = [p_{ij}]

  p_{ij} : probability that the system will go to state j in one step when the system is in state i

  * The same matrix that used to represent the DTMC model of a processor.

- State probability vector of (n+1) step can be represented with the state probability vector of n-th step multiplied by the one-step transition probability matrix.

p(0)P = p(1)
p(1)P = p(2)
⋮
p(n)P = p(n+1)

$$P_j(n+1) = P[X_{n+1} = j]$$
$$= \sum_i P[X_n = i] P[X_{n+1} = j \mid X_n = i]$$

- Chapman-Kolmogorov Equation for DTMC.

$$P_{ij}(n;m) = P\left[X_{n+m} = j \mid X_n = i\right] = \sum_k P_{ik}(n;v) P_{kj}(n+v;m-v)$$

$i, j, k \in S$

$n, m, v \in \mathrm{I}$

- A Markov chain is called (time) homogeneous if

$$P\left[X_{n+k} = j \mid X_n = i\right] = P\left[X_k = j \mid X_0 = i\right] = P_{ij}(0;k) = P_{ij}(k)$$

- Matrix multiplication is associative (not commutative).

$$p(3) = p(2) \cdot P = \left\{p(1) \cdot P\right\} \cdot P = p(0) \cdot P \cdot P \cdot P = p(0) \cdot P^3$$

- N-step transition probability matrix
  - One step transition probability matrix raised to a power n.
  - 현 상태에서 n step후에 system 이 각 state에 있을 확률
  - $\mathrm{P}(n) = \mathrm{P}^n$

- Ergodic MC일때 limiting distribution

$$p(\infty) = \lim_{n \to \infty} p(0) \cdot P^n = \pi$$

가 존재하고 그것은 steady state probability vector이며, p(0)에 무관하게 일정하다.

p(0)	p(1)	p(2)	p(3)	p(4)	...	p(∞)
1	0.20	0.265	0.1805	0.20560	...	0.215554
0	0.75	0.375	0.4350	0.37725	...	0.380389
0	0.00	0.225	0.2025	0.21150	...	0.190194
0	0.05	0.110	0.1170	0.12115	...	0.116653
0	0.00	0.025	0.0650	0.08450	...	0.097210

p(0)	p(1)	p(2)	p(3)	p(4)	...	p(∞)
0	0.30	0.15	0.204	0.1974	...	0.215554
1	0.35	0.48	0.372	0.3900	...	0.380389
0	0.30	0.21	0.228	0.2028	...	0.190194
0	0.10	0.11	0.121	0.1193	...	0.116653
0	0.00	0.05	0.075	0.0905	...	0.097210

Computation of the Steady State Probability Vector

$$p(n) \cdot P = p(n+1) \qquad n \to \infty$$

$$\lim_{n\to\infty} p(n)P = \lim_{n\to\infty} p(n+1)$$

$$\pi P = \pi$$

Solving $\pi P = \pi$ with an extra equation $\sum_{i}^{} \pi_i = 1$ gives π

(Why one more equation? Because $\pi P = \pi$ is not linearly independent)

- 예)

$$[\pi_1,\ \pi_2,\ \pi_3,\ \pi_4,\ \pi_5] \begin{bmatrix} 0.20 & 0.75 & 0.00 & 0.05 & 0.00 \\ 0.30 & 0.30 & 0.30 & 0.10 & 0.00 \\ 0.00 & 0.55 & 0.40 & 0.05 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.50 & 0.50 \\ 0.60 & 0.00 & 0.00 & 0.00 & 0.40 \end{bmatrix} = [\pi_1,\ \pi_2,\ \pi_3,\ \pi_4,\ \pi_5]$$

$$\frac{1}{5}\pi_1+\frac{3}{10}\pi_2+\frac{3}{5}\pi_5=\pi_1$$
$$\frac{3}{4}\pi_1+\frac{3}{10}\pi_2+\frac{11}{20}\pi_3=\pi_2$$
$$\frac{3}{10}\pi_2+\frac{2}{5}\pi_3=\pi_3$$
$$\frac{1}{20}\pi_1+\frac{1}{10}\pi_2+\frac{1}{20}\pi_3+\frac{1}{2}\pi_4=\pi_4$$
$$\frac{1}{2}\pi_4+\frac{2}{5}\pi_5=\pi_5$$

Solve these simultaneous equations for all of the elements of π in terms of one of them. Since the equations are linearly dependent, we can omit one of them, and obtain the following solution.

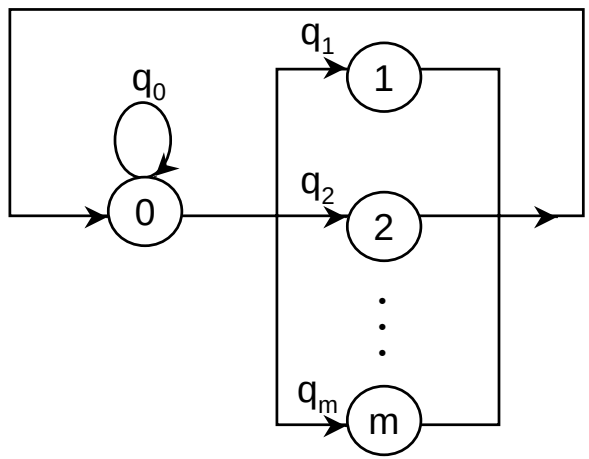
$$\pi_1=\frac{17}{30}\pi_2,\quad \pi_3=\frac{1}{2}\pi_2,\quad \pi_4=\frac{23}{75}\pi_2,\quad \pi_5=\frac{23}{90}\pi_2$$

We can now substitute back into our normalization equation to get the value of π_2 .

$$\left(\frac{17}{30}+1+\frac{1}{2}+\frac{23}{75}+\frac{23}{90}\right)\pi_2=1\quad ,\qquad \pi_2=\frac{450}{1183}$$

- A model of a uniprogrammed computer system with 1 CPU, m I/O devices
  - The system will be in one of (m+1) states :

    state 0 (CPU) or state i (I/0 on device i)
  - Assume that the request for device i occurs at the end of a CPU burst with probability q_i
  - Assume that a program finishes at the end of a CPU burst with probability, q_o, $\sum_{i=0}^{m} q_i = 1$
  - No gap between programs



$$P = \begin{bmatrix} q_0 & q_1 & q_2 & \cdot & \cdot & \cdot & q_m \\ 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix} \begin{matrix} state & 0 \\ state & 1 \\ state & 2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ state & m \end{matrix}$$

- This DTMC is irreducible, recurrent and aperiodic.
  So it is ergodic and there exist the limiting probability distribution.

$$\pi = \pi p$$

$$\begin{cases} \pi_0 = \pi_0 q_0 + \sum_{j=1}^{m} \pi_j \\ \pi_j = \pi_0 q_j \qquad j = 1, 2, \cdots, m \\ \sum_{j=0}^{m} \pi_j = 1 \end{cases}$$

Solving the above $\pi_0 = \pi_0 \sum_{j=1}^{m} q_j = 1$

Since $\sum_{j=1}^{m} q_j = 1 - q_0$,

$$\pi_0 (1 + 1 - q_0) = 1$$

$$\therefore \ \pi_0 = \frac{1}{2 - q_0}$$

$$\pi_j = \pi_0 \cdot q_j = \frac{q_j}{2 - q_0} \qquad \text{j= 1, 2, } \cdots \text{, m}$$

- In a real time interval T, the number of visits to device j will be $\pi_j T$ on the average.

4. CTMC

- Analysis of CTMC(continuous time Markov chain) is similar to that of DTMC except that the transitions can take place at any instance of time.

$$CTMC \quad \{X(t),\ t \geq 0\}$$

$$\mathrm{P}\left[X(t) = x \,|\, X(t_n) = x_n,\ X(t_{n-1}) = x_{n-1},\ X(t_{n-2}) = x_{n-2},\ \cdots,\ X(t_0) = x_0\right]$$

$$= \mathrm{P}\left[X(t) = x \,|\, X(t_n) = x_n\right] \qquad for \quad t_0 < t_1 < \cdots < t_n < t$$

- The behavior of the Markov chain is characterized by
  1. Probability distribution of the initial state of the system
  2. The transition probability

$$p_{ij}(t_1, t_2) = P\left[X(t_2) = j \mid X(t_1) = i\right] \qquad t_1 < t_2$$

$\rightarrow$ Transition probability matrix $\mathrm{P}(t_1, t_2) = \left[p_{ij}(t_1, t_2)\right]$

$$p_j(t) = P\left[X(t) = j\right] = \sum_i P\left[X(t) = j \mid X(t') = i\right] \cdot P\left[X(t') = i\right]$$

$$= \sum_i p_{ij}(t', t) \cdot p_i(t') \qquad t' < t_2$$

$\rightarrow$ State probability vector $p(t)$

Let $t_2 = t_1 + \Delta t$

$$p(t) \cdot \mathrm{P}(t, t+\Delta t) = p(t+\Delta t)$$

$$p(t) \cdot \mathrm{P}(t, t+\Delta t) - p(t) = p(t+\Delta t) - p(t)$$

$$p(t) \cdot \{\mathrm{P}(t, t+\Delta t) - I\} = p(t+\Delta t) - p(t)$$

devide by Δt, $p(t)\dfrac{\mathrm{P}(t, t+\Delta t) - I}{\Delta t} = \dfrac{p(t+\Delta t) - p(t)}{\Delta t}$ ..... (1)

For a sufficiently small Δt, define an infinitesimal matrix Q(t)

$$\mathrm{Q(t)} \triangleq \lim_{\Delta t \to 0} \frac{\mathrm{P}(t, t+\Delta t) - I}{\Delta t}$$

Then taking a $\Delta t \to 0$, (1) becomes

$$p(t) \lim_{\Delta t \to 0} \frac{\mathrm{P}(t, t+\Delta t) - I}{\Delta t} = \lim_{\Delta t \to 0} \frac{p(t+\Delta t) - p(t)}{\Delta t}$$

$$p(t)Q(t) = \frac{d}{dt} p(t)$$

Transition Rate Matrix

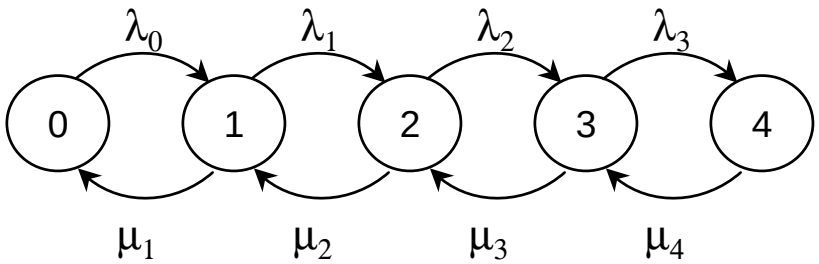
Transition Rate Matrix

- Q(t)
  - Nondiagonal elements : parameters of the exponential probability density function for each state change.
  - Diagonal elements : sum of all the rates for changing from the current state to any other state, but with negative sign.

- If Q(t) is independent of t, it is called time homogeneous.

  Thus we get $p(t)Q = \frac{d}{dt}p(t)$

- Example : birth-death process



$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & 0 \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & 0 \\ 0 & 0 & \mu_3 & -(\lambda_3+\mu_3) & \lambda_3 \\ 0 & 0 & 0 & \mu_4 & -\mu_4 \end{bmatrix} \end{array}$$

- CTMC dose not have self-loop transitions.
- State 1 에서 state 0, state 2로 움직일 수 있는데 state 0로 움직일 확률 : $\frac{\mu_1}{\lambda_1+\mu_1}$

Steady State Probabilities

- Kolmogorov Equations

$$p(t)Q = \frac{d}{dt}p(t)$$

- What is $\frac{d}{dt}p(t)$?
  - Rate of the change of the state probability vector
  - If t $\rightarrow \infty$, the system becomes stable and the rate of change becomes 0.

Let $\pi \triangleq \lim_{t\to\infty} p(t)$ then $p(t)Q = \frac{d}{dt}p(t) \rightarrow \pi Q = 0$

- Example :

  Steady state probabilities of the previous example $(\lambda_i = \lambda,\ \mu_i = \mu)$

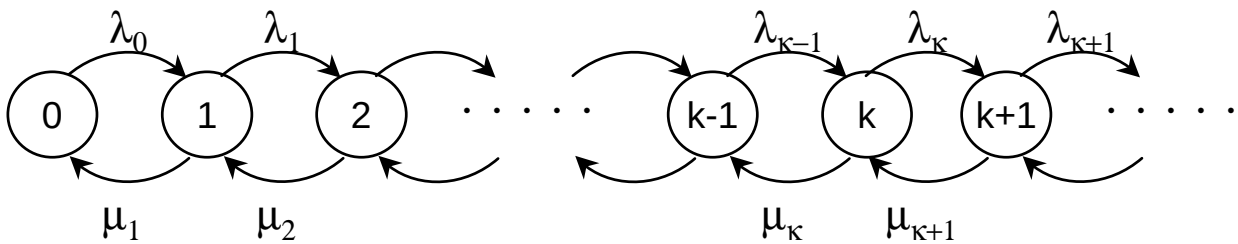
$$[\pi_0,\ \pi_1,\ \pi_2,\ \pi_3,\ \pi_4]\begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & 0 & 0 \\ 0 & \mu & -(\lambda+\mu) & \lambda & 0 \\ 0 & 0 & \mu & -(\lambda+\mu) & \lambda \\ 0 & 0 & 0 & \mu & -\mu \end{bmatrix} = 0$$

$$-\lambda\pi_0 + \mu\pi_1 = 0$$
$$\lambda\pi_0 - (\lambda+\mu)\pi_1 + \mu\pi_2 = 0$$
$$\lambda\pi_1 - (\lambda+\mu)\pi_2 + \mu\pi_3 = 0$$
$$\lambda\pi_2 - (\lambda+\mu)\pi_3 + \mu\pi_4 = 0$$
$$\lambda\pi_3 - \mu\pi_4 = 0$$
$$\pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$$

Balance Equation

Balance Equation

- 각 state별로 그 state로의 inflow와 outflow의 관계를 나타내는 식



$$p(t) = \left(p_0(t),\ p_1(t),\ p_2(t),\ \cdots, p_{k-1}(t),\ p_k(t),\ p_{k+1}(t),\ \cdots\right)$$

$$Q = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ \cdot \\ \cdot \\ k-1 \\ k \\ k+1 \\ \cdot \\ \cdot \end{array} \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & & & & \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & & & & \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & -(\lambda_{k-1}+\mu_{k-1}) & \lambda_{k-1} & 0 & 0 \\ & & & & \mu_k & -(\lambda_k+\mu_k) & \lambda_k & 0 \\ & & & & 0 & \mu_{k+1} & -(\lambda_{k+1}+\mu_{k+1}) & \lambda_{k+1} \\ & & & & & & & \\ & & & & & & & \end{bmatrix}$$

$$\frac{d}{dt}p_k(t) = \lambda_{k-1}\cdot p_{k-1}(t) - (\lambda_k+\mu_k)\cdot p_k(t) + \mu_{k+1}\cdot p_{k+1}(t)$$

$$\frac{d}{dt}p_0(t) = -\lambda_0\cdot p_0(t) + \mu_1\cdot p_1(t)$$

- Steady state에서는 각 상태 확률 $p_k(t)$의 변화율은 0이다.

$$0 = -(\lambda_k + \mu_k)\pi_k + \lambda_{k-1} \cdot \pi_{k-1} + \mu_{k+1} \cdot \pi_{k+1} \quad \text{----①}$$

$$0 = -\lambda_0 \cdot \pi_0 + \mu_1 \cdot \pi_1 \quad \text{----②}$$

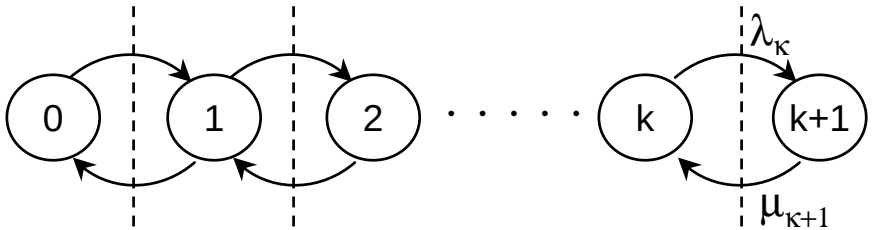
Rates of flow into a state = Rates of flow out of the state

$$\begin{cases} \lambda_k \pi_k + \mu_k \pi_k & \text{: flow out} \\ \lambda_{k-1} \pi_{k-1} + \mu_{k+1} \pi_{k+1} & \text{: flow in} \end{cases}$$

- Balance equation ①, ② 를 정리하면

$$\lambda_\kappa \pi_\kappa - \mu_{\kappa+1}\pi_{\kappa+1} = \lambda_{\kappa-1}\pi_{\kappa-1} - \mu_\kappa \pi_\kappa = \cdots = \lambda_0\pi_0 - \mu_1\pi_1 = 0$$

$$\lambda_0\pi_0 = \mu_1\pi_1$$

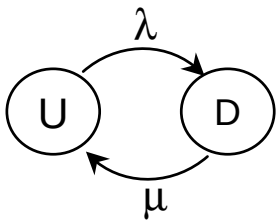


flow rate동일, 즉 flow의 변화 정도는 같아짐

- 즉, $\pi_k = \frac{\lambda_{k-1}}{\mu_k}\pi_{k-1} = \frac{\mu_{k+1}}{\lambda_k}\pi_{k+1}$

- Example : Machine Breakdown

  Consider a component with a constant failure rate λ . Upon failure, it is repaired with an exponential repair-time distribution of parameter μ . $MTTF = \frac{1}{\lambda}$, $MTTR = \frac{1}{\mu}$



Compute the steady-state availability of the component.

$$\pi = (\pi_U,\ \pi_D) \qquad Q = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

From the balance equation,

$$\lambda \pi_U = \mu \pi_D \qquad \therefore \pi_U = \frac{\mu}{\lambda} \pi_D$$

Since $\pi_U + \pi_D = 1$, we get

$$\pi_U + \pi_D = \left(\frac{\mu}{\lambda} + 1\right)\pi_D = 1$$

$$\begin{cases} \pi_D = \dfrac{\lambda}{\lambda + \mu}, \\ \pi_U = \dfrac{\mu}{\lambda + \mu} \end{cases}$$

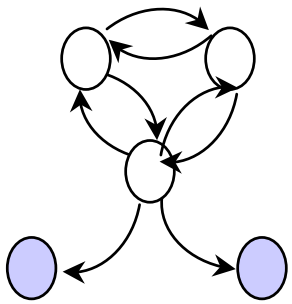
Steady state availability $A = \pi_U$

$$= \frac{\mu}{\lambda + \mu} = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda} + \frac{1}{\mu}} = \frac{MTTF}{MTTF + MTTR}$$

Markov Chains with Absorbing States

Markov chains with absorbing states

- Absorbing states : a state that has incoming transitions only

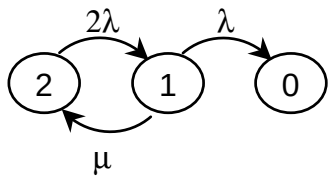


Example

- Assume that we have a two-component parallel redundant system with a single repair facility with rate μ.
- The components have constant failure rate λ .
- When both components have failed, the system is considered to have failed and no recovery is possible.
- What is the reliability of the system?

Solution

- Let the state space be {0, 1, 2} where i represents the number of working components.



- This MC is not ergodic. What we need is transient analysis. Initial state distribution is required for the analysis.
- Assume that $p_2(0) = 1$, $p_1(0) = p_0(0) = 0$.

$$\frac{d}{dt}p_2(t) = -2\lambda p_2(t) + \mu p_1(t)$$
$$\frac{d}{dt}p_1(t) = -(\lambda + \mu)p_1(t) + 2\lambda p_2(t)$$
$$\frac{d}{dt}p_0(t) = \lambda p_1(t)$$

Taking Laplace transform $f(t) \rightarrow \bar{f}(s)$

$$f(t)' \rightarrow s \cdot \bar{f}(s) - f(0)$$

$$s \cdot \overline{p_2}(s) - \underbrace{p_2(0)}_{=1} = -2\lambda \overline{p}_2(s) + \mu \overline{p}_1(s)$$
$$s \cdot \overline{p_1}(s) - p_1(0) = s \cdot \overline{p_1}(s) = -(\lambda + \mu)\overline{p_1}(s) + 2\lambda \overline{p}_2(s)$$
$$s \cdot \overline{p_0}(s) - p_0(0) = s \cdot \overline{p_0}(s) = \lambda \overline{p_1}(s)$$

Solving the above equations for $\overline{p_0}(s)$, we get

$$\overline{p_0}(s) = \frac{2\lambda^2}{s\{s^2 + (3\lambda + \mu)s + 2\lambda^2\}}$$

After an inversion, we get $p_0(t)$, the probability
that no components are operating at time t.
The system is up when 1 or 2 components are operating.

$$\therefore \mathrm{R(t)} = p_2(t) + p_1(t) = 1 - p_0(t)$$

$$MTTF = \int_0^\infty R(t)dt$$

Queueing Theory

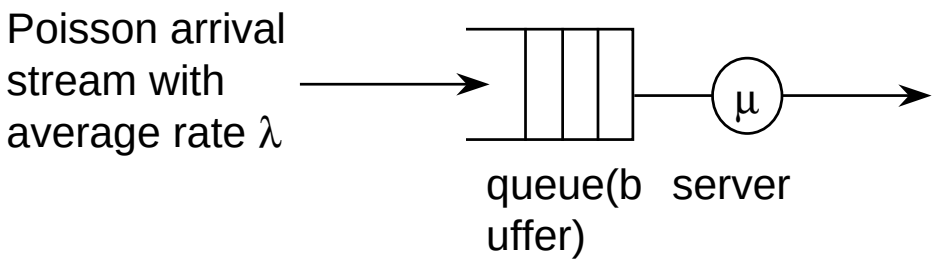
1. Classification of Queues
2. System Utilization
3. Little's Formula
4. M/M/1 Queue
5. M/M/m Queue
6. M/M/∞ Queue
7. M/M/1/L/N Queue

1. Classification of Queues

- Notation of queues is based on five features
  1. The distribution of time between arriving customers
  2. The distribution of time to service a customer
  3. The number of servers
  4. The buffer size that can hold arriving customers
  5. Population size

- Types of time distribution
  - M : Markovian (Memoryless, exponential distribution)
  - D : Deterministic
  - E_r : r-stage Erlang (r개의 연속된 exponential distributed r.v.)
  - H_r : r-stage hyperexponential
  - G : General distribution

- M/M/1/∞/∞
  - Arrival time dist/ service time dist/ no. of server/ buffer size/ pop. size
  - 보통 M/M/1으로 표시



- Service policy

1. FCFS
2. LCFS
3. RR
4. PS (Processor Sharing) : RR with very small time quantum
5. Random
6. Priority
7. Others (SJF, SRTF, ...)

- Priority Scheme
  1. Non-preemptive : Arriving customer does not affect the customer in service
  2. Preemptive-resume :
     - If an arriving customer has higher priority, the current customer in service is preempted
     - When the preempted customer begins service again, the service resumes where it left off
  3. Preemptive-restart
     - If an arriving customer has higher priority, the current customer in service is preempted
     - When the preempted customer begins service again, the service restarts from the beginning as if no service had been received

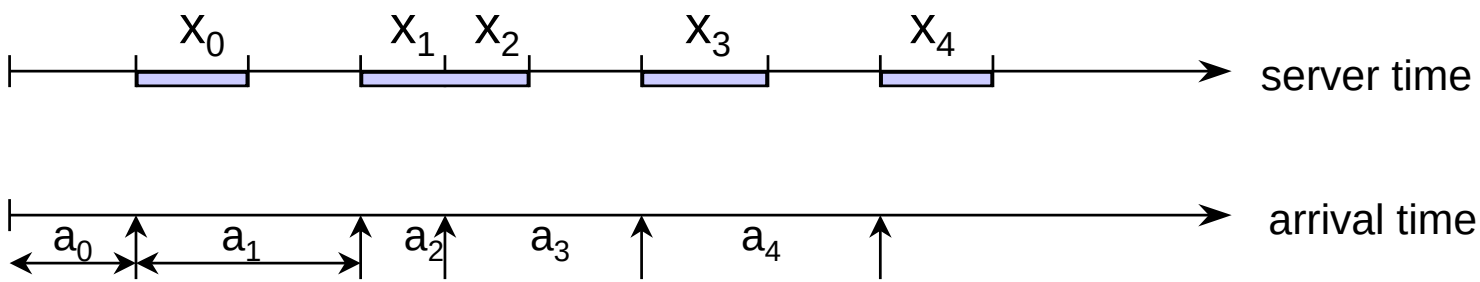
- Because of the memoryless property, preemptive resume and preemptive restart are identical for exponentially distributed service time.

2. System Utilization

- Utilization
  - The fraction of time a system is busy.
  - Increased by the increase of the load on the system
  - A bottleneck will decrease overall utilization

- In a system characterized by a single server queue, the utilization is the fraction of time a server is busy.

  : Sum of the time the system spends serving customer x_i divided by the total amount of time.



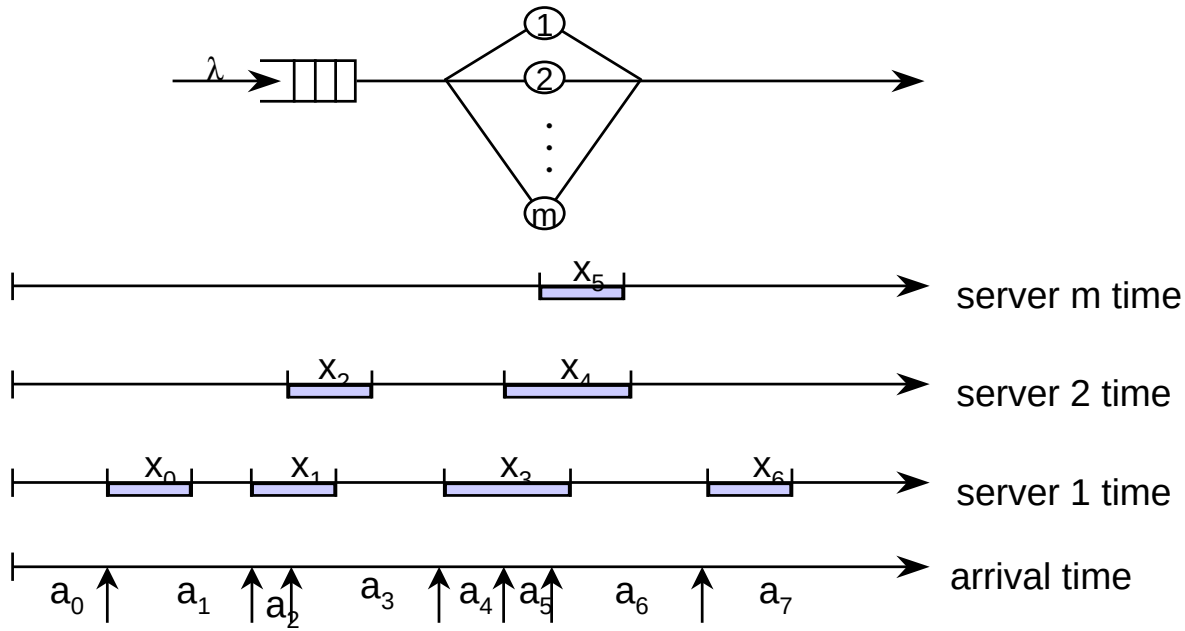
$$\text{Utilization} = \mathrm{P}[\text{system is busy}] = \lim_{N\to\infty} \frac{\sum_{n=0}^{N} x_n}{\sum_{n=0}^{N} a_n} = \lim_{N\to\infty} \frac{\sum_{n=0}^{N} \frac{x_n}{N}}{\sum_{n=0}^{N} \frac{a_n}{N}}$$

$$= \frac{\bar{x}}{\bar{a}} = \frac{\text{mean service time}}{\text{mean interarrival time}}$$

$$= \frac{\frac{1}{\mu}}{\frac{1}{\lambda}} = \frac{\lambda}{\mu} = \frac{\text{arrival rate}}{\text{service rate}}$$

$$\equiv \rho = \lambda \cdot \bar{x} = \lambda \cdot \text{mean service time}$$

Utilization of a multi-server system



- Add up the service times, then divide it by the number of servers to get average

$$P\left[\text{one server is busy}\right] \equiv \rho = \frac{\lambda \bar{x}}{m}$$

3. Little's Formula

- Let a r.v. R denote the response time of a system in the steady state

  W : waiting time of a customer(job)

  S : service time

  N : Number of customers in the queuing system in the steady state

  $$R = W + S$$

  $$E[R] = E[W] + E[S]$$

- Little's Formula (Theorem, Rule)

  $$E[N] = \lambda \times E[R]$$

- Little's Formula holds for a broad variety of queuing systems.
  - Any number of servers, any queuing discipline
    Any arrival, service time distribution
  - Restriction : No job is created or destroyed within the system.

- Example
  - How many people are at McDonald's on the average?
  - Observer at outside can figure it out without entering McDonald's
  - Observe

    32 customers arrive per hour on the average

    each customer exits after 12 minutes on the average after he enters.

$$\lambda = 32/1\,hour = 0.5333/1\,\text{min}utes$$

$$E[R] = 12\,\text{min}utes$$

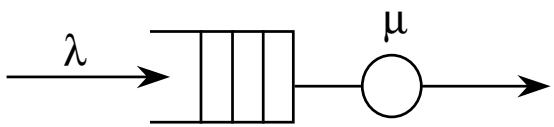
$$E[N] = \lambda \cdot E[R] = 0.5333 \frac{customers}{\text{min}ute} \times 12\,\text{min}ute$$

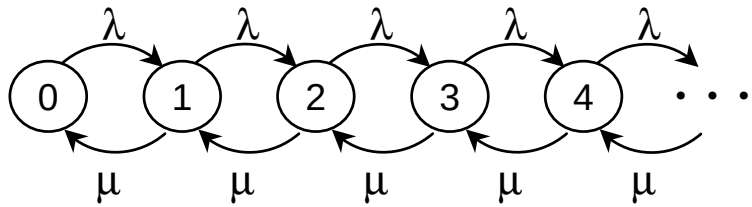
$$= 6.4 \text{ customers}$$

4. M/M/1 Queue

- Example of a Birth-Death process
  - Birth rate is a constant λ
  - Death rate is a constant μ

  for all states





$$\pi_1 = \frac{\lambda}{\mu}\pi_0$$

$$\pi_2 = \frac{\lambda}{\mu}\pi_1 = \left(\frac{\lambda}{\mu}\right)^2 \pi_0$$

.

.

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0$$

- How to compute π_k ?

Using the fact that : $\pi_0 + \sum_{k=1}^{\infty} \pi_k = 1,$

$$\pi_0 + \sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \pi_0 = 1$$

$$\therefore \pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k}$$

If $\mu > \lambda$, we can compute

$$\sum_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k = \frac{\frac{\lambda}{\mu}}{1 - \frac{\lambda}{\mu}} = \frac{\lambda}{\mu - \lambda}$$

therefore $\pi_0 = \frac{1}{1 + \frac{\lambda}{\mu - \lambda}} = \frac{\mu - \lambda}{\mu} = 1 - \frac{\lambda}{\mu}$

Then π_k is $\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0 = \underbrace{\left(\frac{\lambda}{\mu}\right)^k \left(1 - \frac{\lambda}{\mu}\right)}$

Geometric probability density function

- System utilization ρ.

$$\rho = P[k > 0] = 1 - \pi_0 = \frac{\lambda}{\mu}$$

$$\therefore \pi_0 = 1 - \frac{\lambda}{\mu} = 1 - \rho$$

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0 = \rho^k (1 - \rho)$$

❑ Conditions for the ergodicity

① $\lambda < \mu$: system is ergodic

② $\lambda = \mu$: recurrent null

③ $\lambda > \mu$: transient

Condition for stability : $\lambda < \mu$ 즉 $\rho = \frac{\lambda}{\mu} < 1$

❑ Average number of customers in the system

$$\pi_k = (1-\rho)\rho^k$$

Taking z-transform, we get

$$\prod^*(z) = \sum_{k=0}^{\infty} \pi_k \cdot z^k = \sum_{k=0}^{\infty} (1-\rho)(\rho z)^k = \frac{1-\rho}{1-\rho z}$$

$$E[N] = \frac{d}{dz}\prod^*(z)\Big|_{z=1} = \frac{\rho(1-\rho)}{(1-\rho z)^2}\Big|_{z=1} = \frac{\rho}{1-\rho}$$

- Average delay in the system (mean response time)

From the Little's Formula $E[N] = \lambda \cdot E[R]$

$$E[R] = \frac{1}{\lambda} E[N] = \frac{1}{\lambda} \times \frac{\rho}{1 - \rho} = \frac{1}{\mu(1 - \rho)} = \frac{\frac{1}{\mu}}{1 - \rho}$$

- Average number of customers in service

Let a r.v. C have 0 when no customer is in service and have 1 when the server is busy

$$P[C = 1] = \sum_{k=1}^{\infty} \pi_k = \sum_{k=1}^{\infty} (1 - \rho)\rho^k = 1 - \pi_0 = \rho$$

$$E[C] = 0 \cdot P[C = 0] + 1 \cdot P[C = 1] = \rho$$

- Average waiting time in the queue excluding service time

$$E[\text{waiting time before service}] \triangleq W$$

$$= \text{Average delay in the system} - \text{Average service time}$$

$$= E[R] - \frac{1}{\mu}$$

$$= \frac{\frac{1}{\mu}}{1-\rho} - \frac{1}{\mu} = \frac{\rho}{1-\rho} \cdot \frac{1}{\mu} = E[N] \times \text{Average service time}$$

➔ 바로 도착한 customer는 이미 queue에 들어와 있는 E[N]만큼의 customer들이 평균 1/μ 만큼 서비스 받은 후에야 자기가 서비스 받기 시작한다.

- Average number of customers in the queue

$E[\text{Average number in queue}] \triangleq N_q$

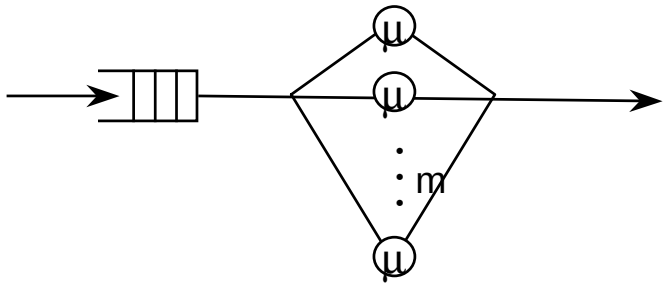
$= \text{Average number in the system} - \text{Average number in service}$

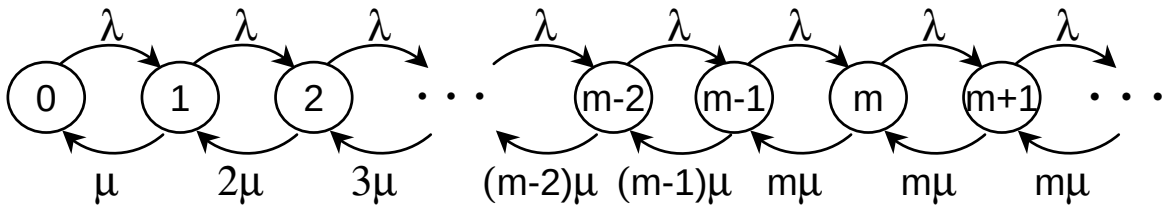
$= E[N] - E[C]$

$= \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$

5. M/M/m Queue

- A queuing system with arrival rate λ and m servers with rate μ each and a common queue





$$\lambda_k = \lambda \qquad k = 0, 1, 2, \cdots$$

$$\mu_k = \begin{cases} k\mu & 0 \le k \le m \\ m\mu & m < k \end{cases}$$

$$\pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} \qquad (k \le m)$$

$$\pi_k = \pi_0 \prod_{i=0}^{m-1} \frac{\lambda}{(i+1)\mu} \prod_{j=m}^{k-1} \frac{\lambda}{m\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{m!\, m^{k-m}} \qquad (k > m)$$

Defining $\rho = \frac{\lambda}{m\mu}$ (the utilization of any individual server)

and using $\sum_{k=0}^{\infty} \pi_k = 1$ we get

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}$$

- Average number of customers in the system

$$E[N] = \sum_{k \geq 0} k \cdot p_k = m\rho + \rho \frac{(m\rho)^m}{m!} \frac{\pi_0}{(1-\rho)^2}$$

- Average number of busy servers

Let a r.v. M denote the number of busy server

$$P[M = k] = \begin{cases} P[N = k] = \pi_k & 0 \leq k \leq m-1 \\ P[N \geq m] = \sum_{k=m}^{\infty} \pi_k = \frac{\pi_m}{1-\rho} & k = m \end{cases}$$

Average number of busy servers,

$$E[M] = \sum_{k=0}^{m-1} k \cdot \pi_k + m \frac{\pi_m}{1-\rho} = \frac{\lambda}{\mu}$$

$$\rho, \text{ utilization of a server} = \frac{\text{Arrival rate to A server}}{\text{Service rate of A server}} = \frac{\frac{\lambda}{m}}{\mu} = \frac{\lambda}{m\mu}$$

$$E[M] = m\rho = \frac{\lambda}{\mu}$$

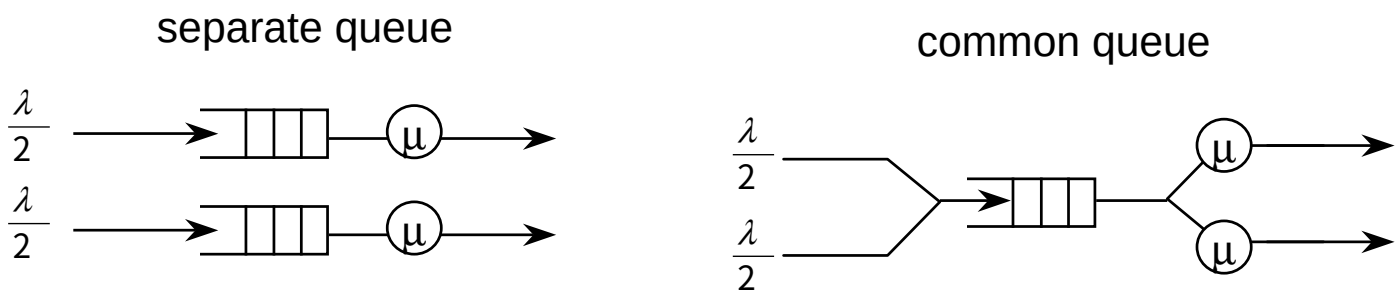
- Probability that an arriving customer is queued (Probability of congestion)

$$P[queuing] = \sum_{k=m}^{\infty} \pi_k = \pi_m + \frac{\lambda}{m\mu}\pi_m + \left(\frac{\lambda}{m\mu}\right)^2 \pi_m + \left(\frac{\lambda}{m\mu}\right)^3 \pi_m + \cdots$$

$$= \sum_{k=m}^{\infty} \rho^{k-m} \cdot \pi_m = \pi_m \sum_{j=0}^{\infty} \rho^j \overset{\leftrightarrow}{=} \frac{\pi_m}{1-\rho}$$

$$= \frac{(m\rho)^m}{m!} \cdot \frac{\pi_0}{1-\rho}$$

Example : Queuing scheme of a multiprocessor OS



Let R_S , R_C be the response time of the separate, common queuing scheme respectively.

Compare $E[R_S]$ and $E[R_C]$

Separate system : 2 independent M/M/1 queue with $\rho = \frac{\lambda}{2\mu}$

Using the fact that $E[N_s] = \frac{\frac{\lambda}{2\mu}}{1-\frac{\lambda}{2\mu}} \quad \left(= \frac{\rho}{1-\rho}\right)$ and Little's Formula,

we get $E[R_s] = \frac{1}{\frac{\lambda}{2}} \cdot E[N_s] = \frac{2}{2\mu - \lambda}$

- Common queue : M/M/2 system

$$E[N_c] = 2\rho + \frac{\rho(2\rho)^2}{2!}\frac{\pi_0}{(1-\rho)^2} \qquad \rho = \frac{\lambda}{2\mu}$$

$$\pi_0 = \left[1 + 2\rho + \frac{(2\rho)^2}{2!}\cdot\frac{1}{1-\rho}\right]^{-1} = \frac{1-\rho}{1+\rho}$$

Thus

$$E[N_c] = \frac{2\rho}{1-\rho^2}$$

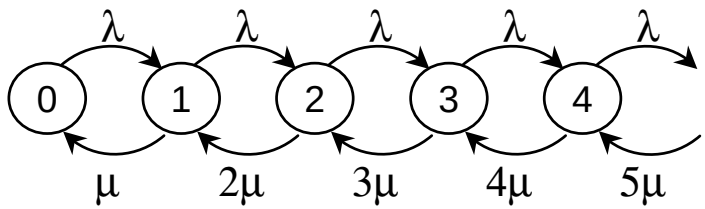
$$E[R_c] = \frac{1}{\lambda}E[N_c] = \frac{4\mu}{4\mu^2 - \lambda^2}$$

$$E[R_s] = \frac{2}{2\mu - \lambda} = \frac{4\mu + 2\lambda}{4\mu^2 - \lambda^2} > E[R_c]$$

$\therefore$ Common queue is better

6. M/M/∞ Queue

- Infinite server queue : no queuing
- A birth-death process with birth rate λ, but death rate is $k \times \mu$ where k is the number of customers in the system.



$$\pi_k = \left(\prod_{j=1}^{k} \frac{\lambda}{j \times \mu} \right) \pi_0 = \left(\frac{\lambda}{\mu} \right) \left(\frac{\lambda}{2\mu} \right) \left(\frac{\lambda}{3\mu} \right) \cdots \left(\frac{\lambda}{k\mu} \right) \pi_0$$

$$= \frac{1}{\mathrm{k!}} \left(\frac{\lambda}{\mu} \right)^k \pi_0$$

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \frac{1}{\mathrm{k!}} \left(\frac{\lambda}{\mu} \right)^k} = \frac{1}{\frac{\left(\frac{\lambda}{\mu} \right)^0}{0!} + \sum_{k=1}^{\infty} \frac{\left(\frac{\lambda}{\mu} \right)^k}{\mathrm{k!}}} = \frac{1}{\sum_{k=0}^{\infty} \frac{\left(\frac{\lambda}{\mu} \right)^k}{\mathrm{k!}}}$$

$$= \frac{1}{e^{\frac{\lambda}{\mu}}} = e^{-\frac{\lambda}{\mu}}$$

$$\pi_k == \frac{1}{\mathrm{k!}} \left(\frac{\lambda}{\mu} \right)^k e^{-\frac{\lambda}{\mu}}$$

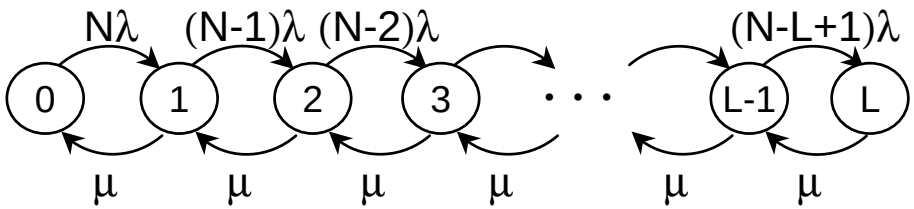
7. M/M/1/L/N Queue

- M/M/1 Queue with a finite amount of storage (L) and a finite population (N)
- In the case of finite population, it is assumed that each individual attempts to join the queue as an independent Poisson process with parameter λ.

  (If there are n individuals, the aggregate arrival rate is n λ)
- Assume L<N. (If L≥N, the queues is equivalent to M/M/1/∞/N Queue)

- Arrival rate

$$\lambda_k = \begin{cases} (N-k)\lambda & k < L \\ 0 & k \geq L \end{cases}$$



$$\pi_k = \left(\prod_{j=0}^{k-1} \frac{(N-j)\lambda}{\mu} \right) \pi_0 = \frac{N!}{(N-k)!} \left(\frac{\lambda}{\mu} \right)^k \pi_0 \qquad (1 \leq k \leq L)$$

$$\pi_k = 0 \qquad (k > L)$$

$$\pi_0 = \frac{1}{1 + \sum_{j=1}^{L} \frac{N!}{(N-j)!} \left(\frac{\lambda}{\mu}\right)^j}$$

$$\pi_k = \frac{\frac{N!}{(N-k)!} \left(\frac{\lambda}{\mu}\right)^k}{1 + \sum_{j=1}^{L} \frac{N!}{(N-j)!} \left(\frac{\lambda}{\mu}\right)^j} = \frac{\frac{N!}{(N-k)!} \left(\frac{\lambda}{\mu}\right)^k}{\sum_{j=0}^{L} \frac{N!}{(N-j)!} \left(\frac{\lambda}{\mu}\right)^j} = \frac{\frac{\left(\frac{\lambda}{\mu}\right)^k}{(N-k)!}}{\sum_{j=0}^{L} \frac{\left(\frac{\lambda}{\mu}\right)^j}{(N-j)!}}$$

Networks of Queues

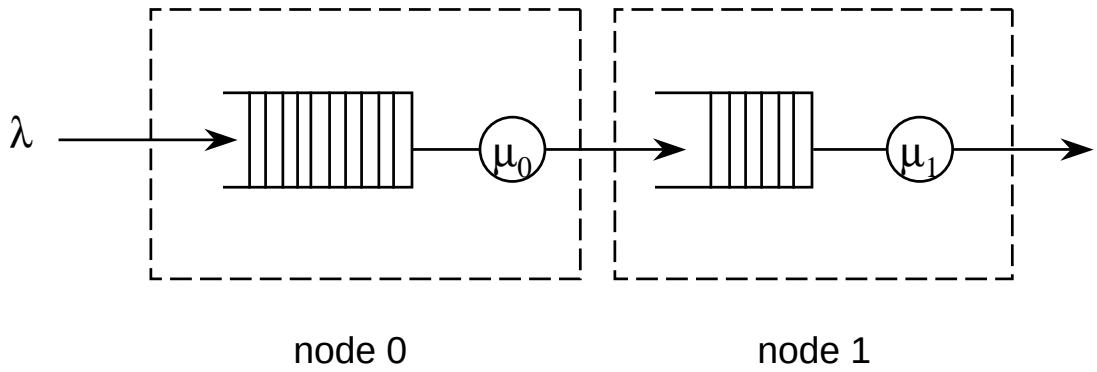
1. Tandem Queues
2. Open Queuing Networks
3. Closed Queuing Network
4. General Network Model

1. Tandem Queues

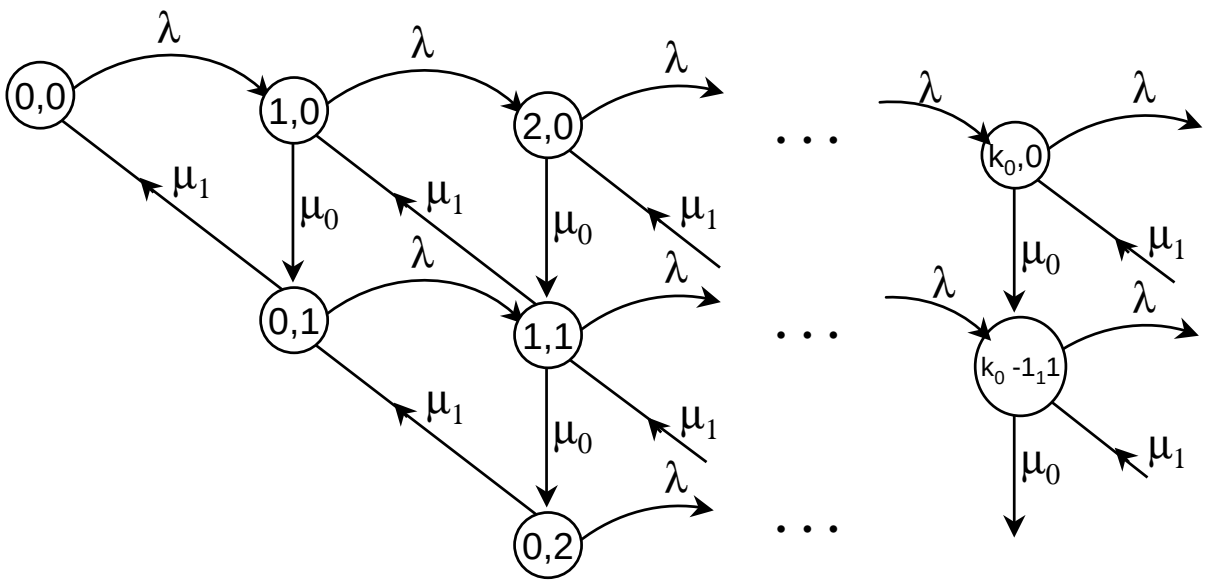
- Two types of networks : Open and Closed
  - An open queuing network is characterized by one or more *sources* of job arrivals and correspondingly one or more *sinks* that absorb jobs departing from the network.
  - In a closed queuing network, jobs neither enter nor depart from the network.
- The behavior of jobs within the network is characterized by
  - the distribution of job service times at each center
  - the probabilities of transitions between service centers
- For each center the number of servers, the scheduling discipline, and the size of the queue must be specified.
  - For an open network, a characterization of job-arrival processes is needed.
  - For a closed network, the number of jobs in the network must be specified.

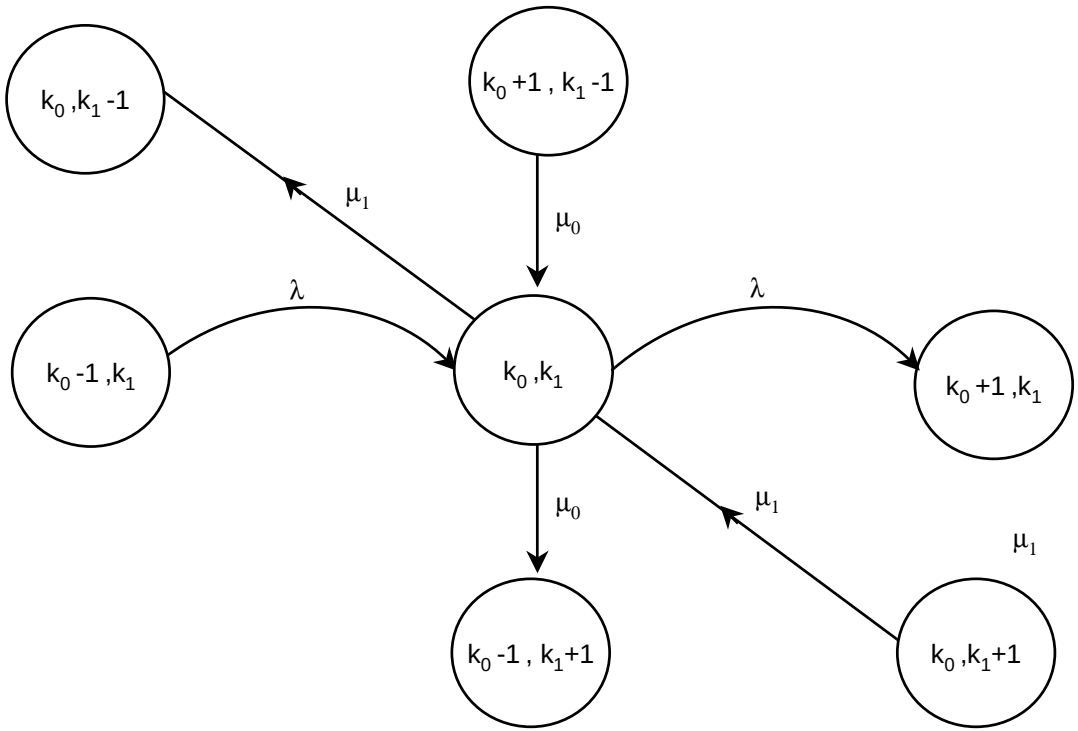
The two-stage tandem network

- The service-time distribution at both nodes is exponential and the arrival process to the node labeled 0 is Poisson.



- The changes of state occur upon a completion of service at one of the two servers or upon an external arrival.
- Since all inter event times are exponentially distributed (by our assumptions), the underlying stochastic process is a Markov chain with the state diagram shown in the following.




$k_0, k_1 - 1$
$k_0 + 1, k_1 - 1$
μ_1
μ_0
λ
λ
$k_0 - 1, k_1$
k_0, k_1
$k_0 + 1, k_1$
μ_0
μ_1
μ_1
$k_0 - 1, k_1 + 1$
$k_0, k_1 + 1$

- Let $p(k_0, k_1)$ be the joint probability of k_0 jobs at node 0 and k_1 jobs at node 1 in the steady state.
  - Equating the rates of flow into and out of the state, we obtain the following balance equations :

$$\begin{aligned}(\mu_0 + \mu_1 + \lambda)\, p(k_0, k_1) &= \mu_0 p(k_0 + 1, k_1 - 1) + \mu_1 p(k_0, k_1 + 1) \\ &\quad + \lambda p(k_0 - 1, k_1), \qquad k_0 > 0, \quad k_1 > 0\end{aligned}$$

  - For the boundary states, we have:

$$\begin{aligned}(\mu_0 + \lambda)\, p(k_0, 0) &= \mu_1 p(k_0, 1) + \lambda p(k_0 - 1, 0), \qquad k_0 > 0, \\ (\mu_1 + \lambda)\, p(0, k_1) &= \mu_0 p(1, k_1 - 1) + \mu_1 p(0, k_1 + 1), \quad k_1 > 0, \\ \lambda p(0, 0) &= \mu_1 p(0, 1)\end{aligned}$$

- The normalization is provided by :

$$\sum_{k_0 \geq 0} \sum_{k_1 \geq 0} p(k_0, k_1) = 1$$

- Therefore the solution of the above relations is:

$$p(k_0, k_1) = (1 - \rho_0)\rho_0^{k_0}(1 - \rho_1)\rho_1^{k_1}, \qquad (1)$$

where $\rho_0 = \lambda / \mu_0$ and $\rho_1 = \lambda / \mu_1$.

- The condition for stability of the system is that both ρ_0 and ρ_1 are less than unity.

- The node labeled 0 is an M/M/1 queue.

$$P(N_0 = k_0) = p_0(k_0) = (1-\rho_0)\rho_0^{k_0}.$$

  - **Burke's theorem**[1956] says that **the departure process of an M/M/1, M/M/c, or M/M/∞ queue is also Poisson with rate λ.**
  - Therefore, the second queue is also an M/M/1 queue.

$$P(N_1 = k_1) = p_1(k_1) = (1-\rho_1)\rho_1^{k_1}.$$

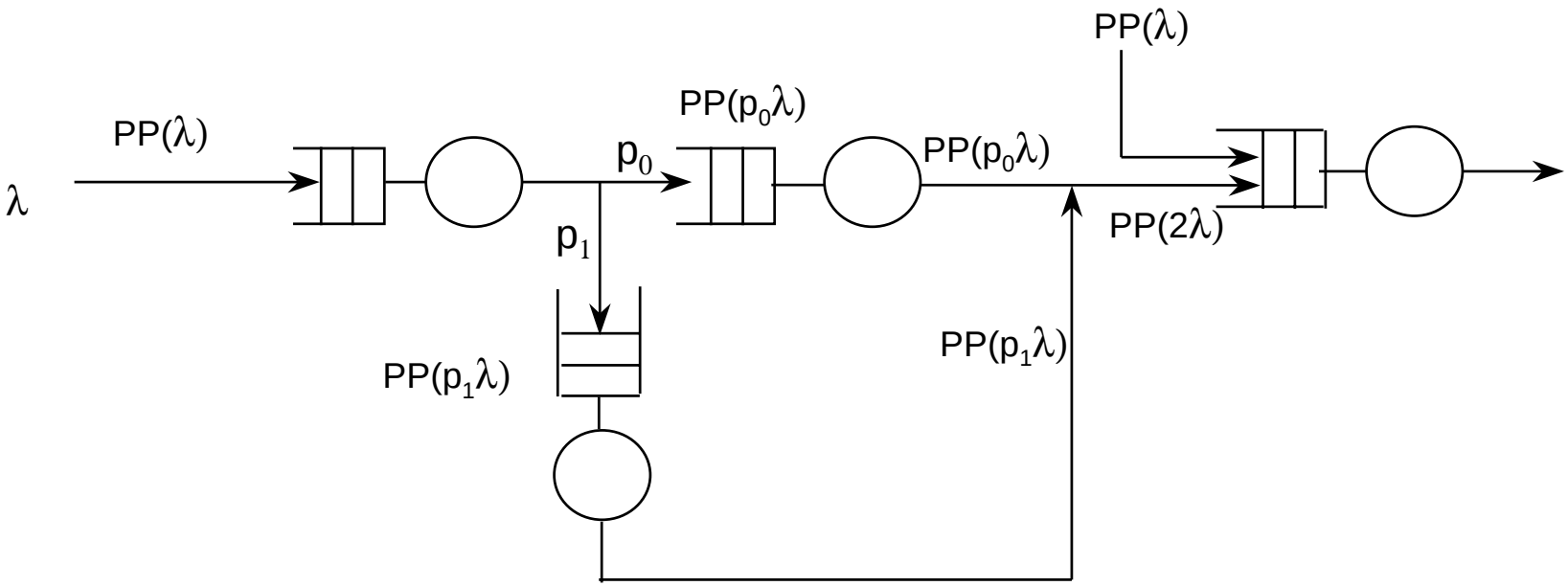
- The solution in Equation (1) becomes:

$$p(k_0, k_1) = (1-\rho_0)\rho_0^{k_0}(1-\rho_1)\rho_1^{k_1} = p_0(k_0)\ p_1(k_1).$$

  - This is the product form solution meaning the product of the marginal probabilities $p_0(k_0)$ $p_1(k_1)$; hence random variables $N_0(t)$ and $N_1(t)$ are independent at the same time instance. (Are they independent random processes? NO!)

  - This product form solution can be generalized to an *m*-stage tandem queue.

- Property of Poisson process
  - Probabilistic splits and probabilistic merge also form Poisson process.



2. Open Queuing Networks

- If a queue is visited more than once(there is a feedback), then the arrival process to that queue will not be Poisson.
- The product-form solution of tandem queues can be generalized to any feed-forward network of exponential queues that is fed from independent Poisson sources.

- Jackson's theorem [1957]
  - The numbers of customers in the queues at time *t* are independent random variables.
  - The steady state probabilities of the individual queues are those of an M/M/c (M/M/1, M/M/∞) system.
  - Let the steady state pmf of the queueing network be *N(t).*

    For any possible state $n = (n_1, n_2, n_3, \ldots n_K)$

$$P[N(t) = n] = P[N_1(t) = n_1] \times P[N_2(t) = n_2] \times P[N_3(t) = n_3] \times \ldots \times P[N_K(t) = n_K]$$

- Consider a simple model of a computer system
  - Jackson's theorem says that two queues will behave like independent M/M/1 queues, so

$$p(k_0, k_1) = (1-\rho_0)\,\rho_0^{k_0}\,(1-\rho_1)\,\rho_1^{k_1}, \qquad (2)$$

$$\rho_0 = \lambda_0 / \mu_0, \;\; \rho_1 = \lambda_1 / \mu_1$$

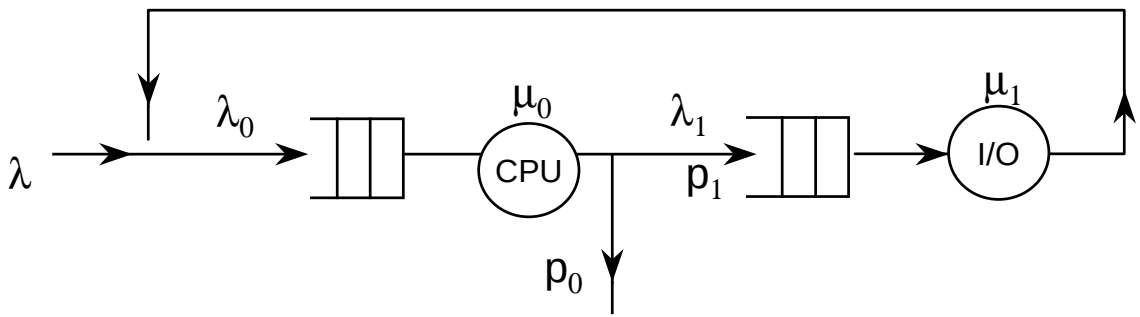


Fig. 1 A model of a computer system

- To apply this result, we have to compute the average arrival rates λ_0 and λ_1 into the two nodes :

$$\lambda_0 = \lambda + \lambda_1$$

$$\lambda_1 = \lambda_0 p_1$$

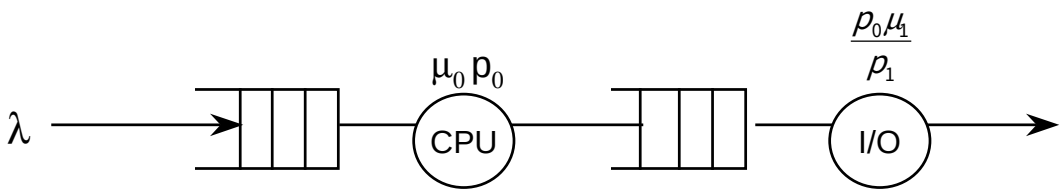
- Thus :

$$\lambda_0 = \frac{\lambda}{1-p_1} = \frac{\lambda}{p_0} \quad \text{and} \quad \lambda_1 = \frac{p_1\lambda}{p_0}$$

$$\rho_0 = \frac{\lambda}{p_0\mu_0} \quad \text{and} \quad \rho_1 = \frac{p_1\lambda}{p_0\mu_1}$$

$$E[N_0] = \frac{\rho_0}{(1-\rho_0)}, \quad E[N_1] = \frac{\rho_1}{(1-\rho_1)}$$

- If we let B_0 denote the total CPU service requirement of a program, then $E[B_0]=1/(p_0 \mu_0)$, the mean service time at the equivalent server. Similarly, $E[B_1]= p_1/(p_0 \mu_1)$.
  - If $\rho_0 > \rho_1$, then $E[B_0] > E[B_1]$. This means the CPU is the bottleneck, the system is said to be CPU-bound.
  - Similarly, if $\rho_0 < \rho_1$, then the system is I/O-bound.



(c) The same "equivalent" network without feedback

- The average response time may be computed by summing the average number of jobs the two nodes and then using Little's formula:

$$E[R] = \left(\frac{\rho_0}{1-\rho_0} + \frac{\rho_1}{1-\rho_1}\right)\frac{1}{\lambda} = \frac{\left(\frac{1}{p_0\mu_0}\right)}{1-\lambda\left(\frac{1}{p_0\mu_0}\right)} + \frac{\left(\frac{p_1}{p_0\mu_1}\right)}{1-\lambda\left(\frac{p_1}{p_0\mu_1}\right)}$$

$$= \frac{E[B_0]}{1-\lambda E[B_0]} + \frac{E[B_1]}{1-\lambda E[B_1]} \qquad (3)$$

- We know that a single queue has the following relation.

$$E[R] = \frac{1}{\lambda}E[N] = \frac{1}{\lambda}\cdot\frac{\rho}{1-\rho} = \frac{\frac{1}{\mu}}{1-\rho} = \frac{\frac{1}{\mu}}{1-\lambda\left(\frac{1}{\mu}\right)}$$

- Therefore Equation (3) gives the average turnaround time of the tandem network without feedback shown in Fig 1 (b).
- To determine E[R] and E[N], it is sufficient to know only the aggregate resource requirements of a job. Details of the pattern of the resource usage are not important for computing these average values.
- Figure 1(a) and 1(b) does not hold with respect to the distribution function $F_R(x)$ of the response time.

- Jackson's result applies in even greater generality.
- Requirement of Jacksonian product form solution
  - The queueing network has the external Poisson arrival process.
  - Each queue has FCFS service policy.
  - Service times of each queue are independent each other.

→ Even if **each** queue of the open queuing system does not have Poisson process for input and output, we can still have product form solution.

- Consider an open queuing network with (m+1) nodes, where the ith node consists of c_i exponential servers each with mean service time of $1/\mu_1$ seconds.
  - Jackson's theorem states that each node behaves like an independent M/M/c_i queue and, therefore, the steady-state probability of k_i customers at node i, i = 0, 1, ..., m is given by the product from:

$$p(k_0, k_1, ... , k_m) = p_0(k_0) .... p_m(k_m),$$

  - where $p_i(k_i)$ is the steady-state probability of finding k_i jobs in an M/M/c_i queue with input λ_i and with average service time $1/\mu_1$ for each of the c_i servers.

3. Closed Queuing Network

- Let us now assume a large value of λ, so that the probability that there is at least one customer in the job scheduler queue is very high.
- We may then assume that the departure of a job immediately triggers the scheduling of an already waiting job into main memory.
- Consider a cyclic queuing model shown in Fig. 2.
  - Represent the state of the system by a pair, (k_0, k_1) where k_i denotes the number of jobs at node i (i = 0, 1).
  - $k_0 + k_1 = n$ is the degree of multiprogramming.
  - The dot pattern on the line $k_0 + k_1 = n$ of Fig. 3 is the finite-state space of the cyclic (closed) queuing network.

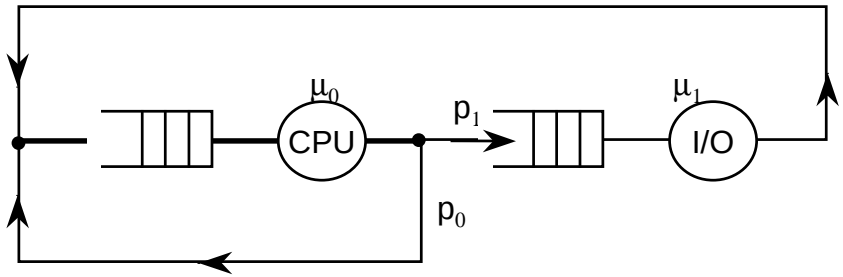


Fig. 2 The closed cyclic queuing model

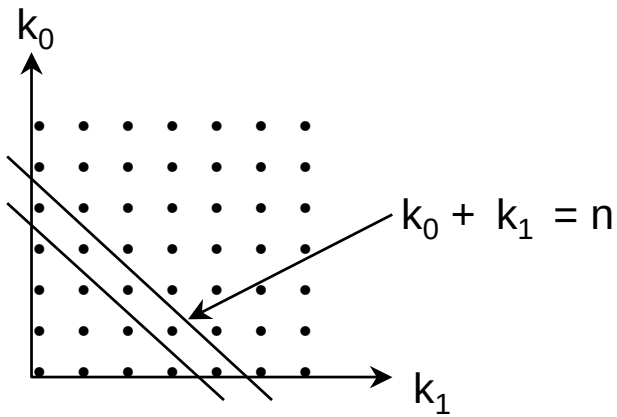
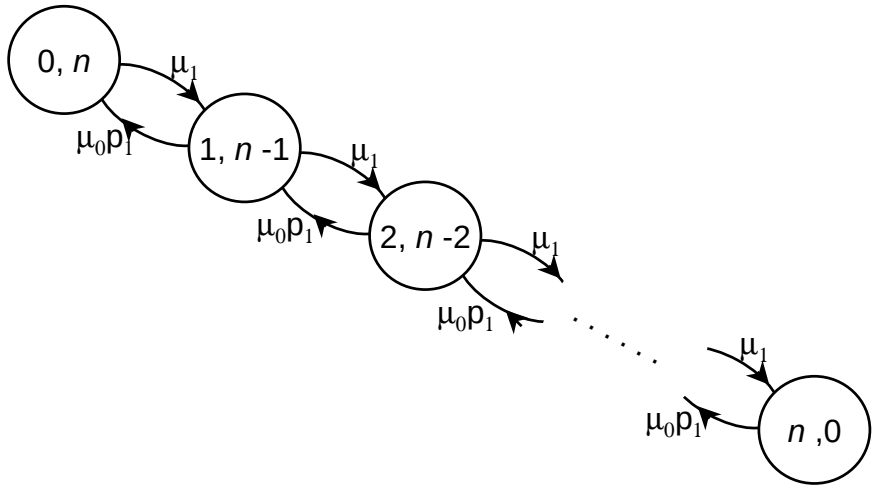


Fig. 3 State spaces for two-stage networks

- The state diagram for the closed cyclic queuing model



- The balance equations are given by:

$$\begin{aligned}
(\mu_1 + \mu_0 p_1)\, p(k_0, k_1) &= \mu_0 p_1 p(k_0 + 1, k_1 - 1) + \mu_1 p(k_0 - 1, k_1 + 1), \quad k_0,\ k_1 > 0, \\
\mu_1 p(0, n) &= \mu_0 p_1 p(1, n - 1), \\
\mu_0 p_1 p(n, 0) &= \mu_1 p(n - 1, 1).
\end{aligned}$$

- If we let $\rho_0 = a/\mu_0$ and $\rho_1 = a p_1/\mu_1$, where a is an arbitrary constant, that the steady-state probability $p(k_0, k_1)$: (by Gordon & Newell)

$$p(k_0, k_1) = \frac{1}{C(n)} \rho_0^{k_0} \rho_1^{k_1} .$$

  - The normalizing constant C(n) is chosen so that :

$$\sum_{\substack{k_0 + k_1 = n \\ k_0, k_1 \geq 0}} p(k_0, k_1) = 1.$$

  - The choice of the constant a is quite arbitrary in that the value of $p(k_0, k_1)$ will not change with a, although the intermediate values ρ_0, ρ_1, C(n) will depend upon a.
  - Two popular choices of the constant a are $a = 1$ and $a = \mu_0$.
  - Choosing $a = \mu_0$ we have $\rho_0 = 1$ and $\rho_1 = \mu_0 p_1/\mu_1$.

$$p(k_0, k_1) = \frac{1}{C(n)} \rho_1^{k_1} .$$

- Using the normalization condition, we get :

$$1 = \frac{1}{C(n)} \sum_{k_1=0}^{n} \rho_1^{k_1} = \frac{1}{C(n)} \frac{1-\rho_1^{n+1}}{1-\rho_1}$$

$$C(n) = \begin{cases} \dfrac{1-\rho_1^{n+1}}{1-\rho_1}, & \rho_1 \neq 1 \\ n+1, & \rho_1 = 1. \end{cases}$$

- How to get C(n)?

$$C(n) = \sum_{A} \prod_{i=0}^{m} \rho_i^{k_i} \qquad A : \sum k_i = n, k_i \geq 0.$$

  - By direct evaluation
  - By using generating function

- Now the CPU utilization U_0 may be expressed as :

$$U_0 = 1 - p(0,n) = 1 - \frac{\rho_1^{\,n}}{C(n)},$$

$$U_0 = \begin{cases} \dfrac{\rho_1 - \rho_1^{\,n+1}}{1 - \rho_1^{\,n+1}}, & \rho_1 \neq 1 \\ \dfrac{n}{n+1}, & \rho_1 = 1. \end{cases}$$

- The average throughput is given by:

$$E[T] = \mu_0 p_0 U_0$$

4. General Network Model

- More general queuing network that have been found to have a product-form solution.
  - Chandy, Howard, Towsley 1977, Chandy and Yeh 1978, Lam 1977, Kleinrock 1975, Kobayashi 1978, ...
  - Any differentiable service-time distribution can be allowed at a node, provided that the scheduling discipline at the node is PS (processor sharing) or LCFS-PR.
  - Any differentiable service-time distribution can also be allowed at a node with ample servers. (no queuing needed)

  `
  - Networks with multiple job types can be analyzed.

Stochastic Petri Nets

1. 성능분석 기법 개요
2. 페트리넷 정의
3. Petri Net 모델 해석

1. 성능분석 기법 개요

- Non state-space model
  - System component 개별적 해석결과의 조합
  - Inter-dependency, concurrency, resource contention 표현에 부적합
  - Product form queueing network
  - Series-parallel graphs
  - Fault-tree
  - Reliability graph
- State-space model
  - State machine
  - Markov models
  - Petri nets

Why Markov Models ?

- 장점:
  - Interdependencies among system components
  - Concurrency
  - Synchronization
  - Contention for resources
  - For both the performance and reliability analysis
  - Generalization to Markov Reward models
- 단점:
  - Largeness problem
  - Exponential assumption

Markov model 단점의 보완책

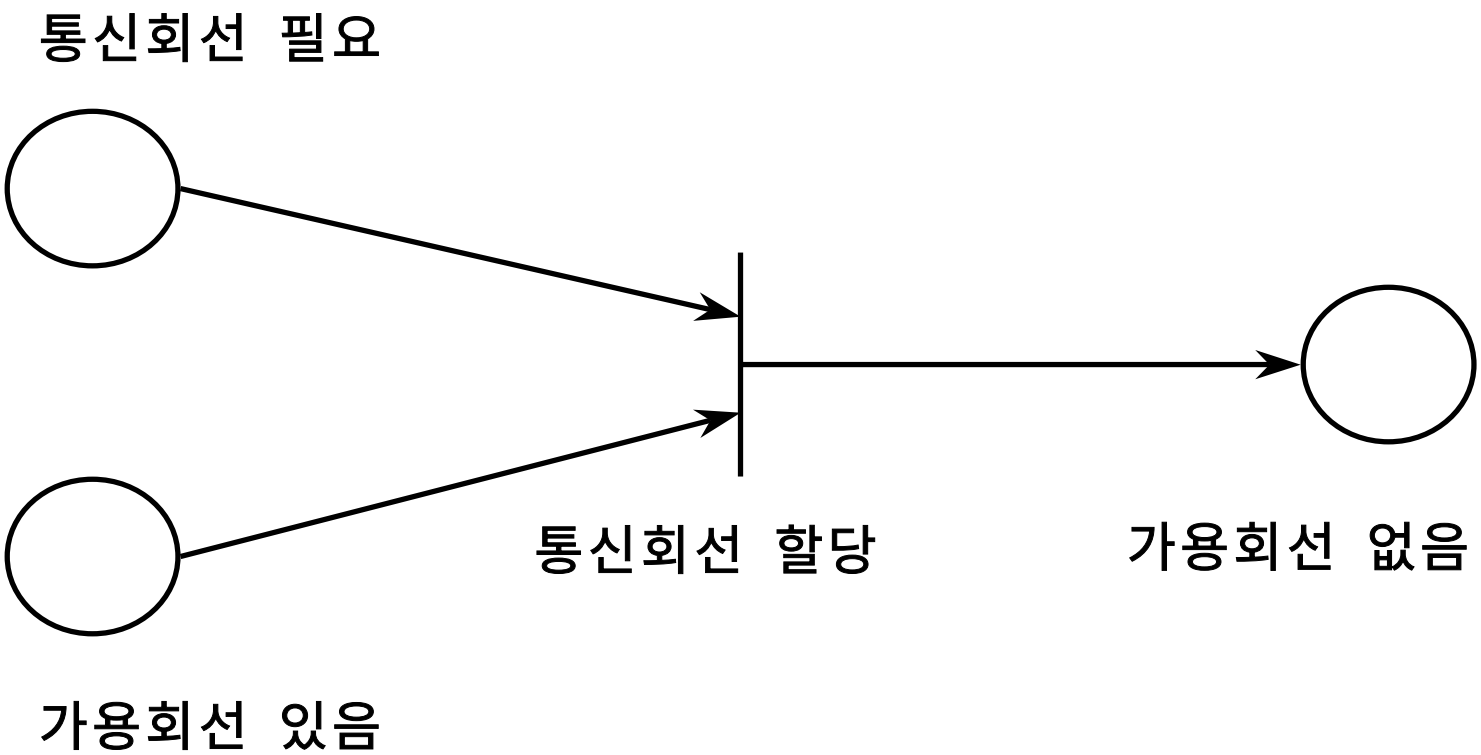
- Largeness problem
  - Automatic state space generation : **Stochastic Petri nets**, high level languages
  - Efficient numerical solution technique
  - Model decomposition
  - Model approximation(simplification)
- Exponential assumption
  - Non-homogeneous Markov process
  - Semi-Markov process
  - Markov regenerative process
  - Approximation of non-exponential distributions by PH-type expansion
    - Largeness problem
  - Discrete-event simulation
- Needs
  - Stochastic Petri nets for
  - (Non-exponential event times with)
  - Analytic solution methods

2. 페트리넷 정의

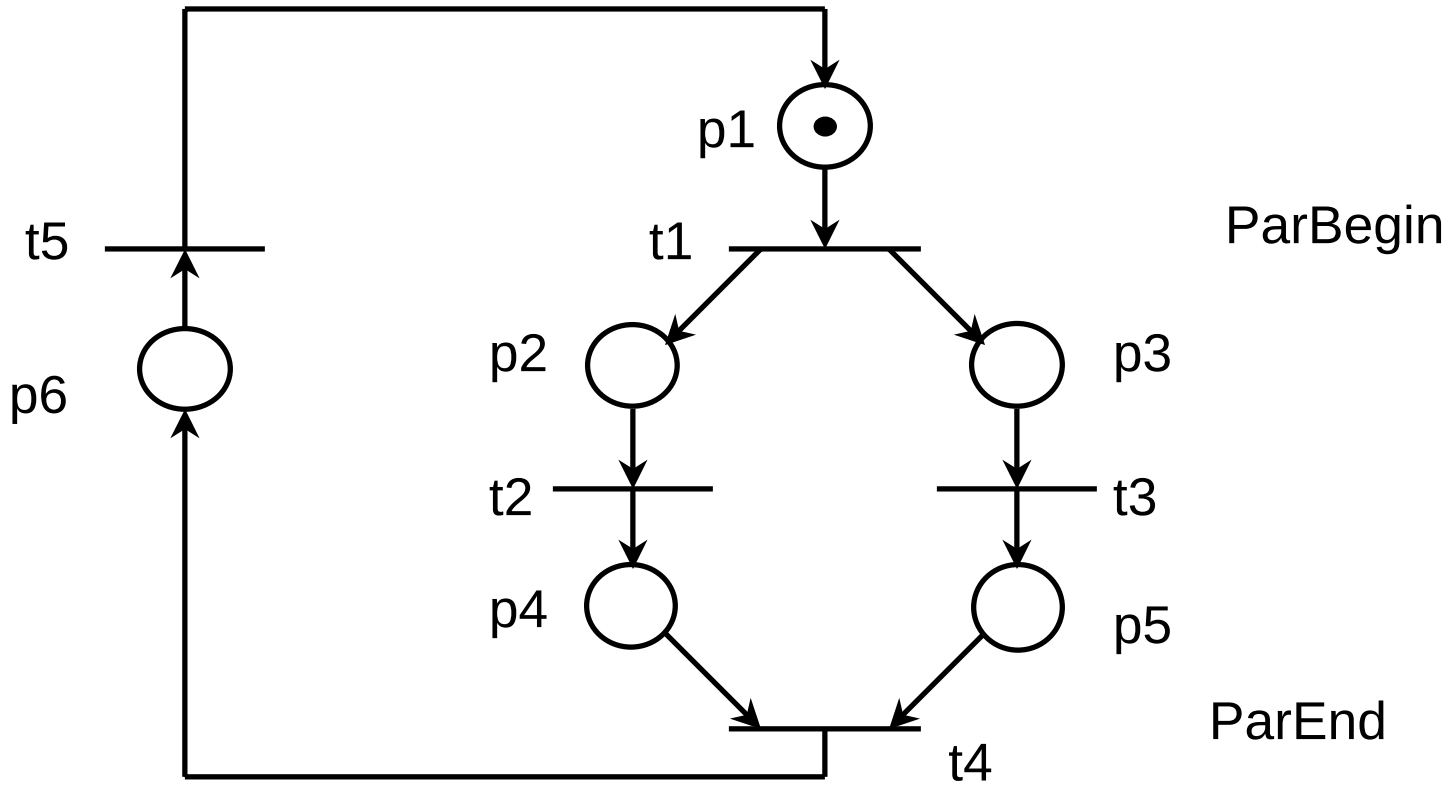
- "Abstract and Formal Model of Information Flow"
- "Bipartite Directed Graphs Consisting of Places and Transitions"

- Petri Nets, PN=(P,T,A,C,M0)
  - P = {p1, p2, p3, ... , pj}, a finite set of places
  - T = {t1, t2, t3, ... ,tm}, a finite set of transitions
  - $A \subseteq \{P \times T\} \cup \{T \times P\}$, a finite set of arcs
  - $C : A \rightarrow \{1, 2, 3, ...\}$, a function of arc multiplicity
  - $M0 : P \rightarrow \{0, 1, 2, 3,...\}$, the initial marking

페트리넷의 예(1)



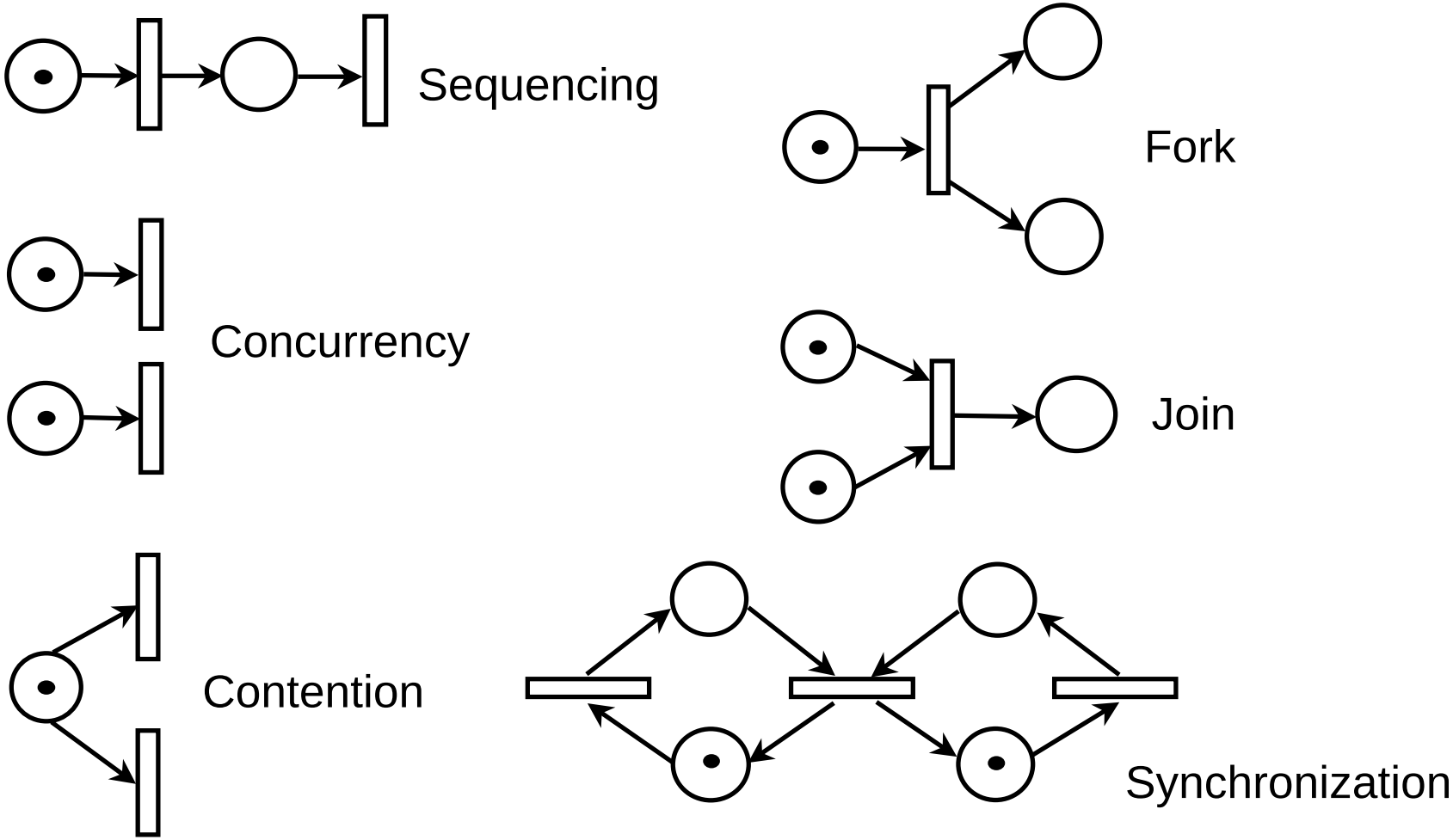
페트리넷의 예(2)



페트리넷 모델링의 특징

- Sequential execution(ordering)
- Concurrency, parallelism
- Synchronization
- Asynchronous activities
- Contention
- Non determinism
- Interdependency
- Hierarchical modeling

동작정의, 순서 및 실시간성 모델링



페트리넷과 시간개념

- 페트리넷에 시간 개념 표현
- 필요성
  - Dynamic System Scheduling
  - Race Condition Modeling
  - Timing in Communication Protocols
  - Performance, Reliability Evaluation

페트리넷 성능분석 방법

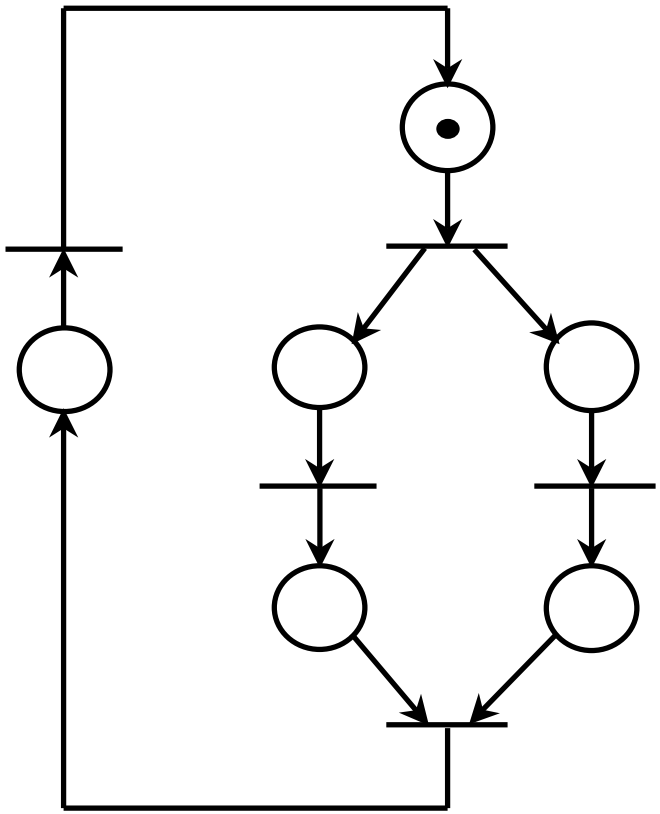
1. 시스템의 동작에 따른 상태변화를 페트리넷으로 모델링

2. 상태변화에 소요되는 시간 지연을 transition에 표현

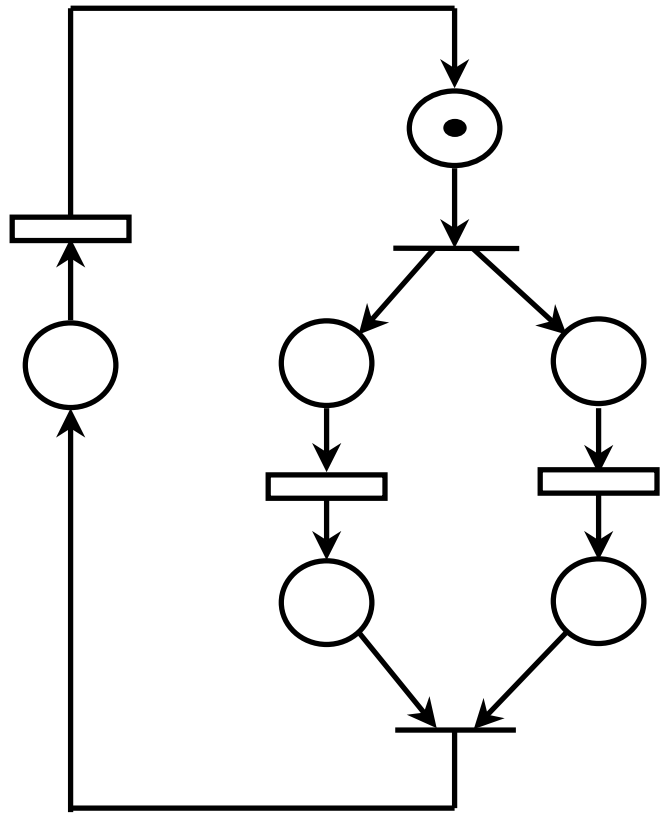
3. 페트리넷 모델을 Markov model로 변환
   (1) Petri nets ⟶ Reachability graph
   (2) Reachability graph ⟶ Reduced reachability graph

4. Markov model을 해석
   (1) Solve Markov model : state probabilities
   (2) Compute performance metrics with reward parameters

시간 지연의 모델링



페트리넷

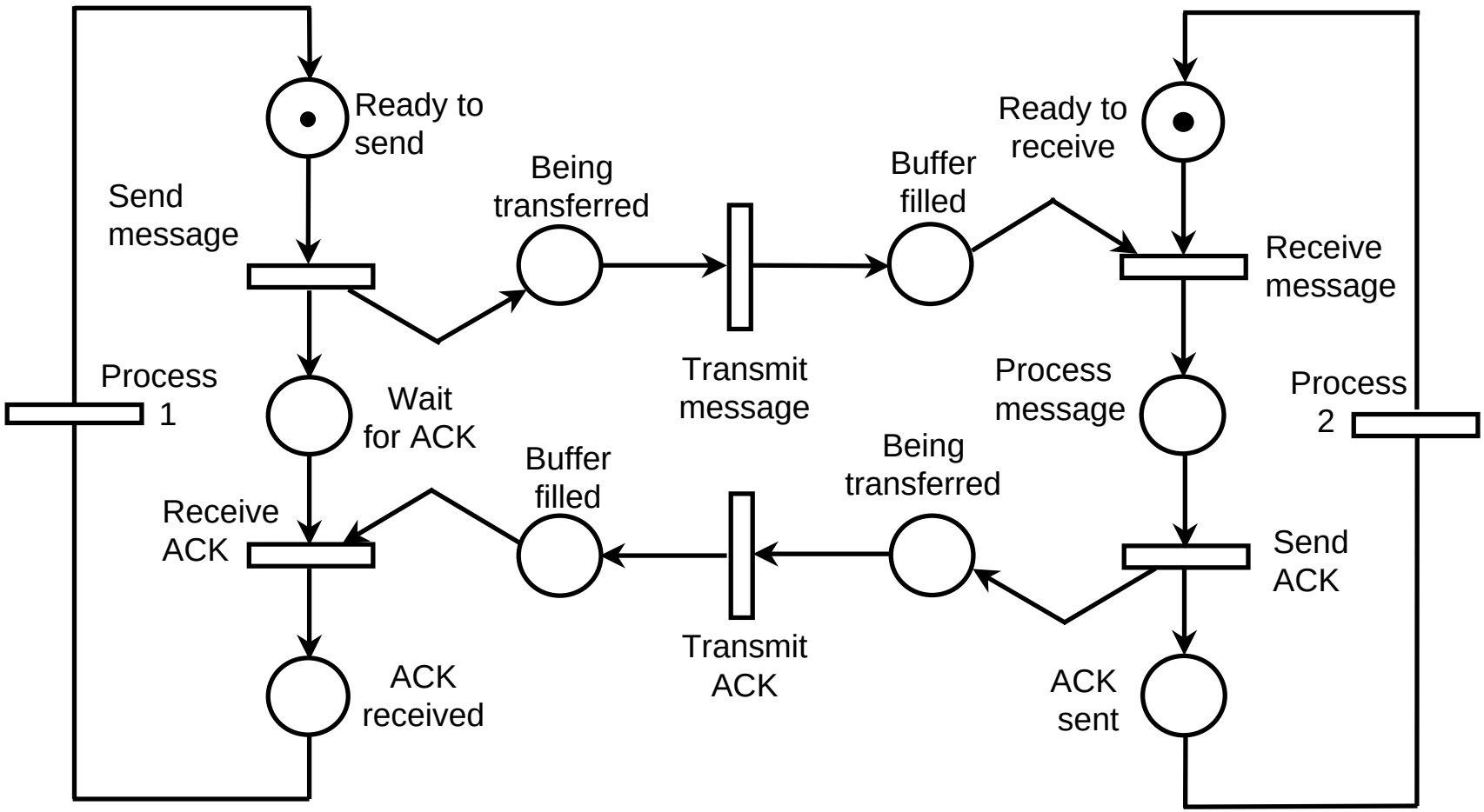


시간 페트리넷

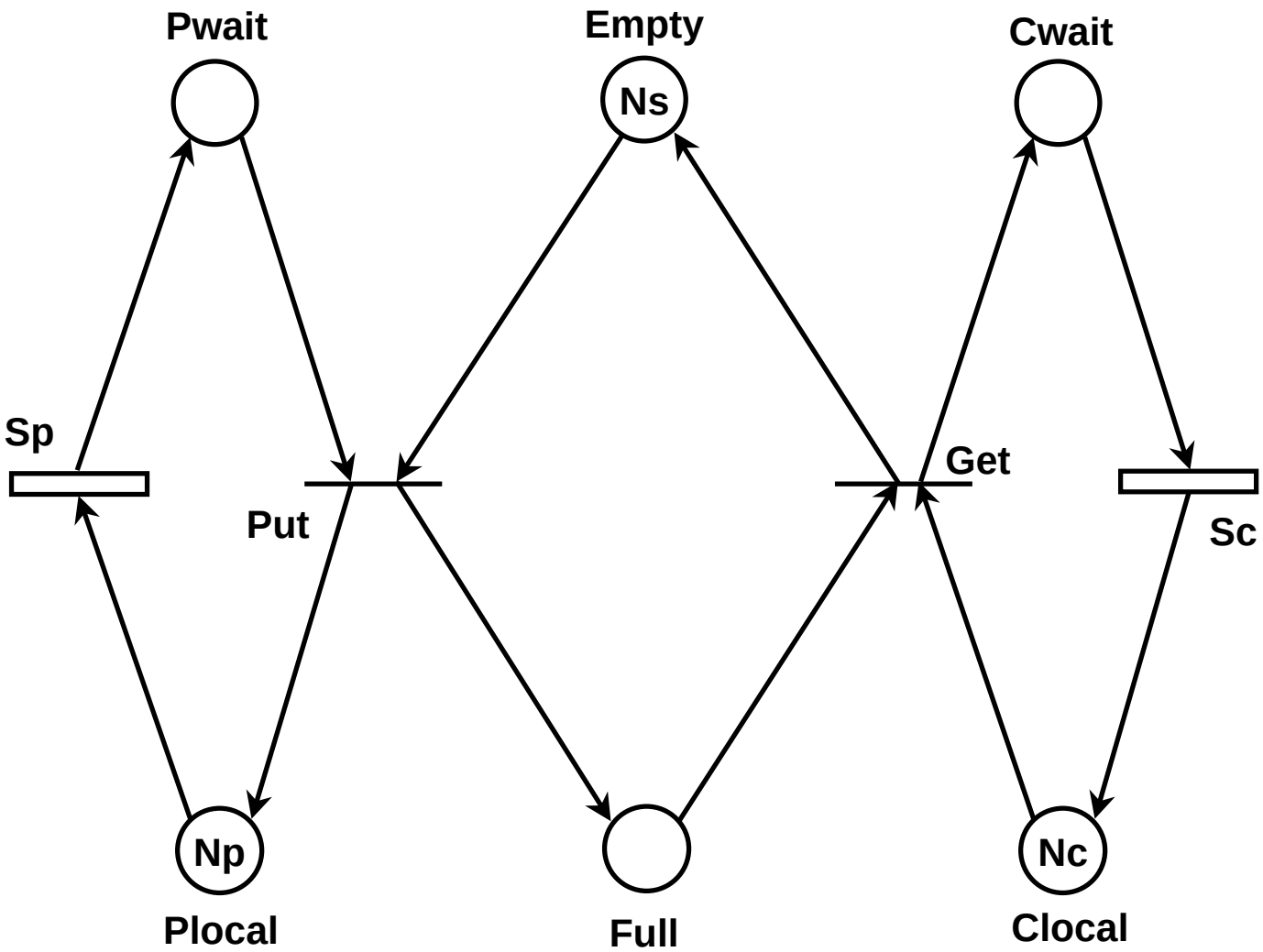
시간 페트리넷과 Markov Models

- Deterministic Timed Nets : (Generalized) Timed Petri Nets
- Stochastic Petri Nets(SPN) → CTMC
- Generalized SPN(GSPN) → CTMC
- Discrete Time SPN → DTMC
  - EXP(GEO) distributed firing times
- Extended SPN(ESPN) → SMP
  - EXP and GEN distributed firing times
  - Reduced reachability graph must have Markov property each time when marking change occurs.
- Deterministic & SPN(DSPN) → MRGP
  - EXP and DET distributed firing times
  - At most one DET transition can be enabled in a marking.
- Markov Regenerative SPN(MRSPN) → MRGP
  - EXP and GEN distributed firing times
  - At most one GEN transition can be enabled in a marking.

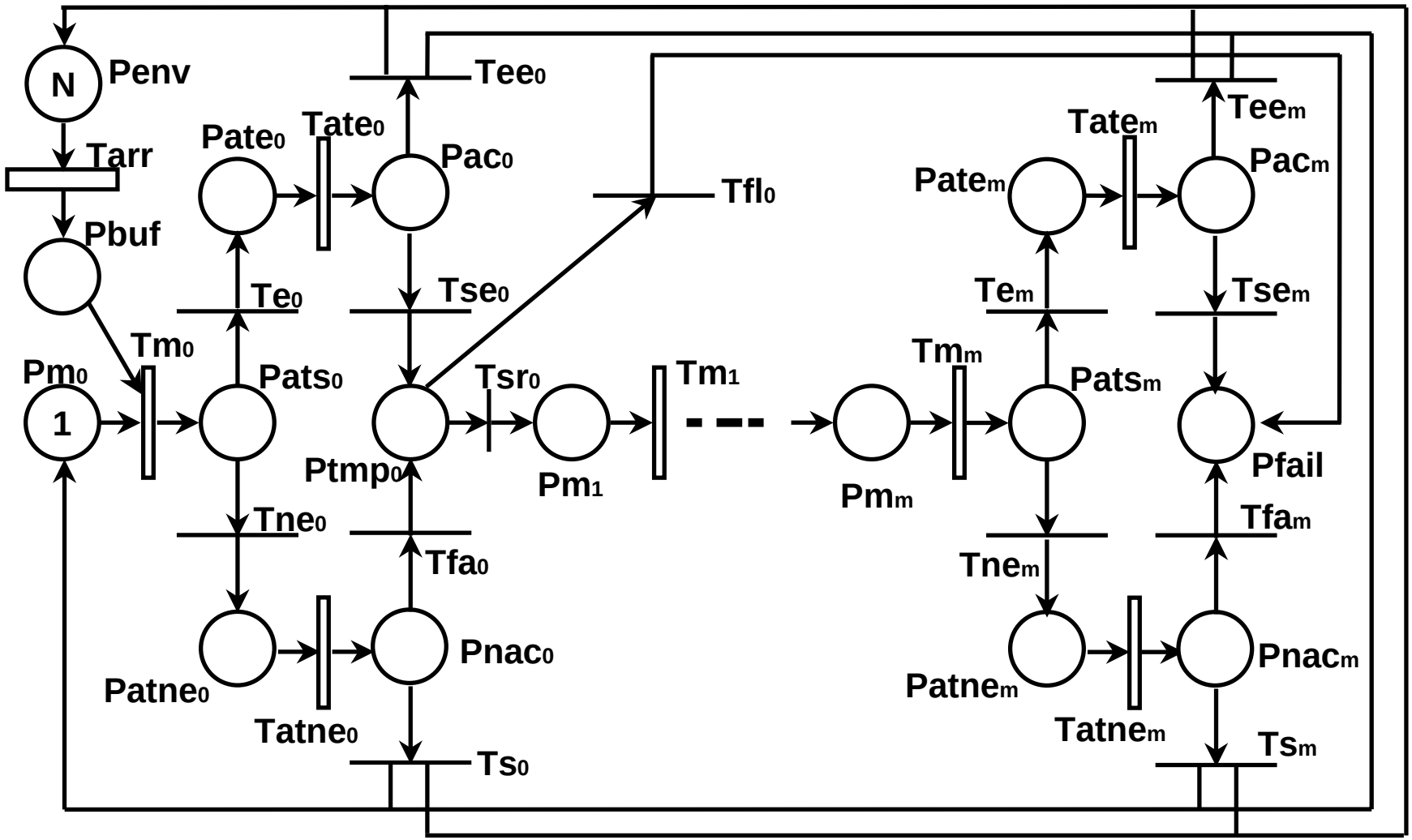
통신 프로토콜의 페트리넷 모델



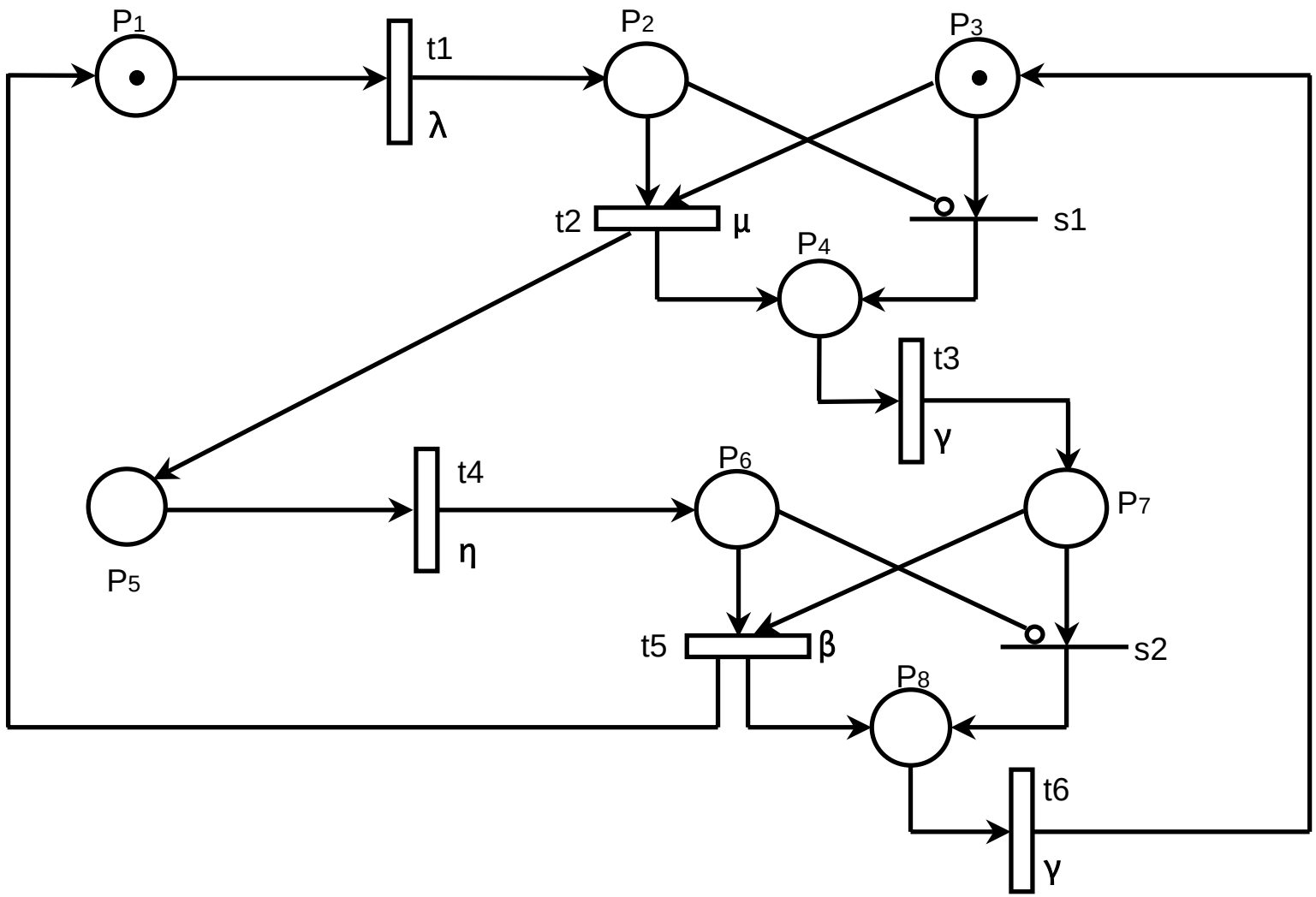
Producer-Consumer System Model



Fault-Tolerant SW Model



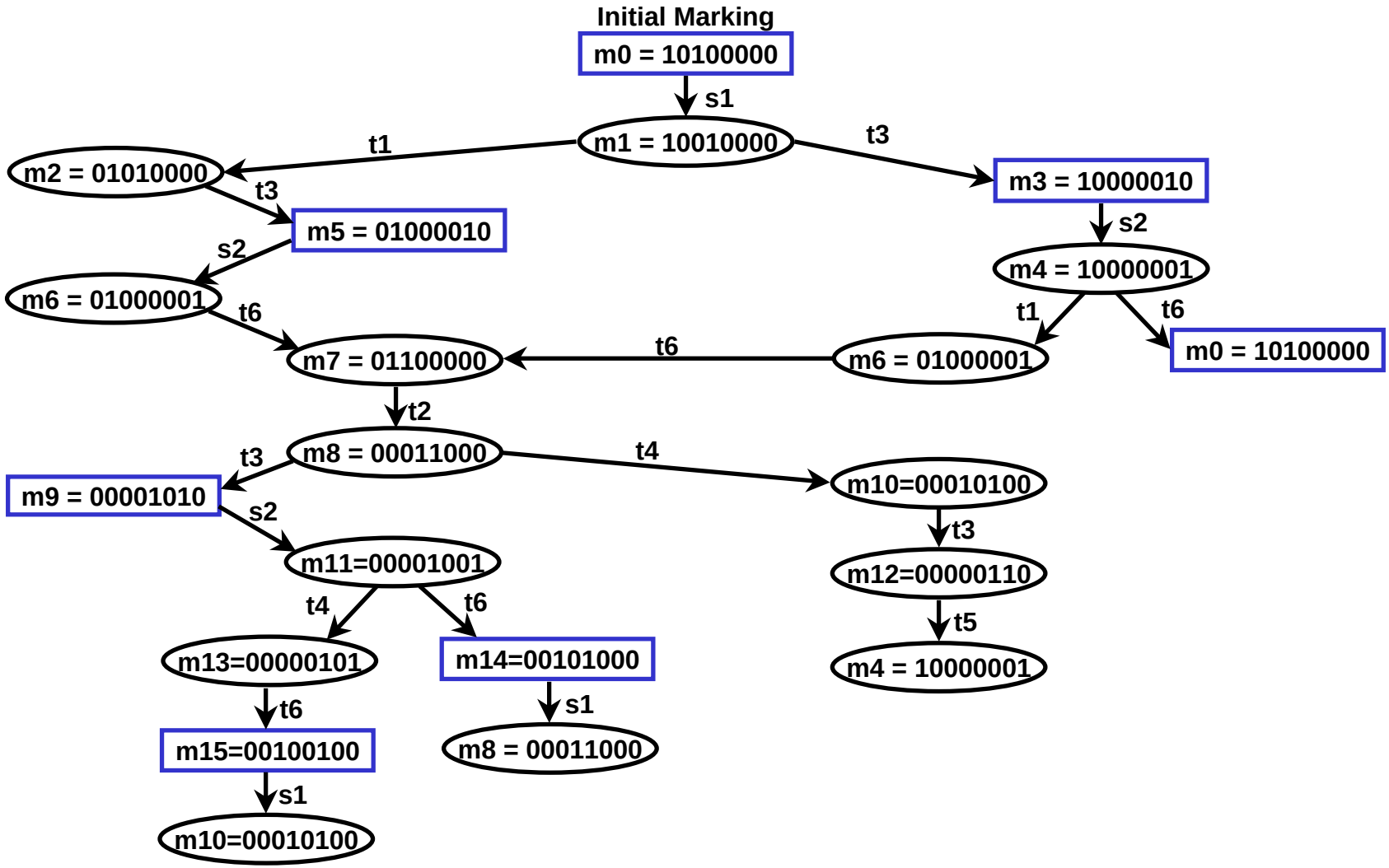
Model of a Simple Client-Server System



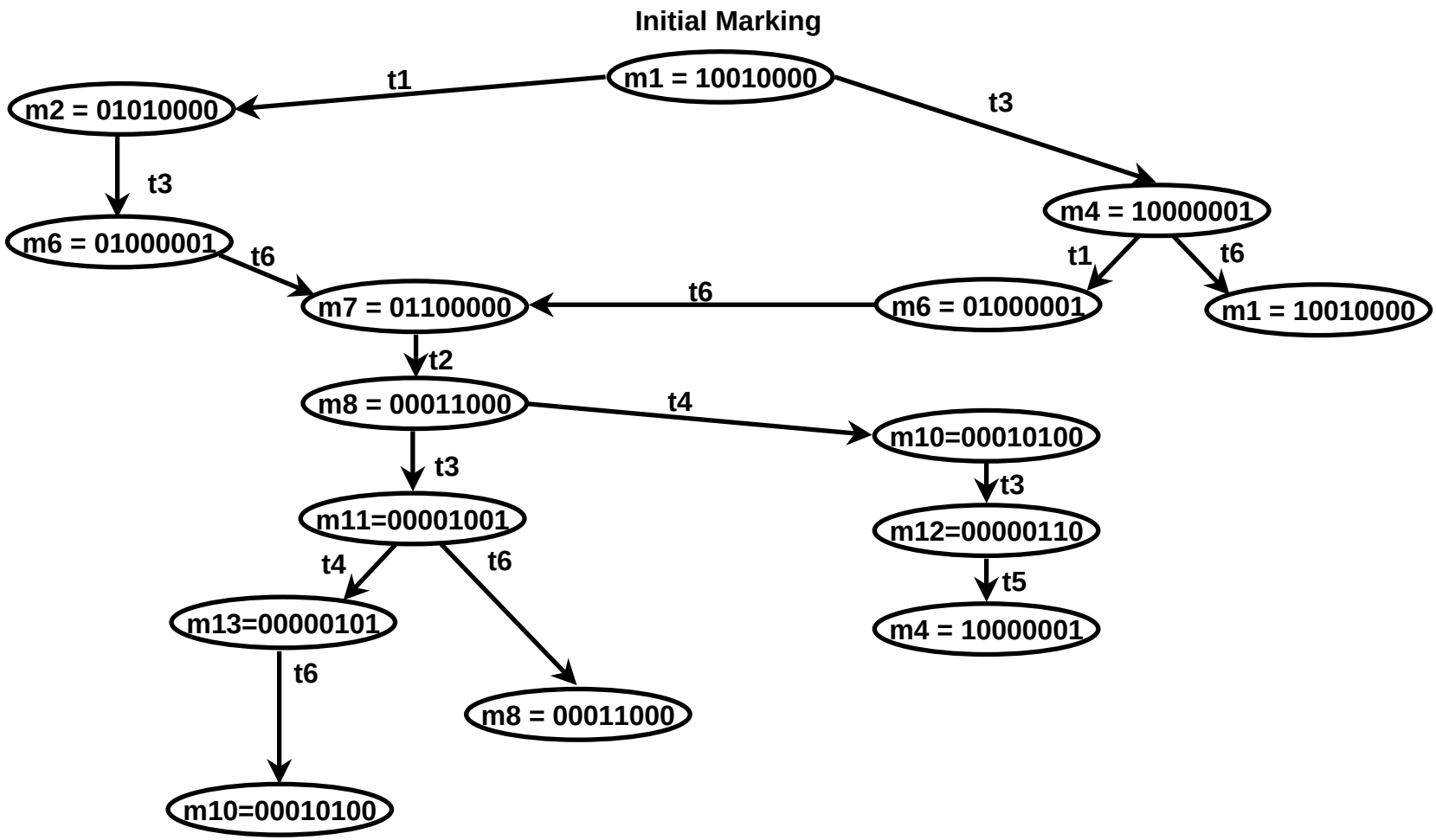
3. Petri Nets 모델 해석

- Marking : 페트리넷 내의 token 분포 상태
  - 각 place 별 token 수의 나열
  - 페트리넷으로 표현된 **시스템의 상태**를 의미
  - (1 0 2 0) : 1번 place 에 token 1 개, 2번 place 에 token 0 개, 3번 place 에 token 2 개, 4번 place 에 token 0 개가 배치된 **상태**
- Tangible marking
  - Timed transition 만 enable 된 marking
  - 이 marking 에서 후속 marking 으로 이동하기까지 시간 소요
- Vanishing marking
  - 적어도 한 개의 immediate transition이 enable 된 marking
  - 이 상태에서 지체되는 시간 없음 (시간 지연 = 0)
  - 후속 tangible marking에 합침 (probabilistic switching)

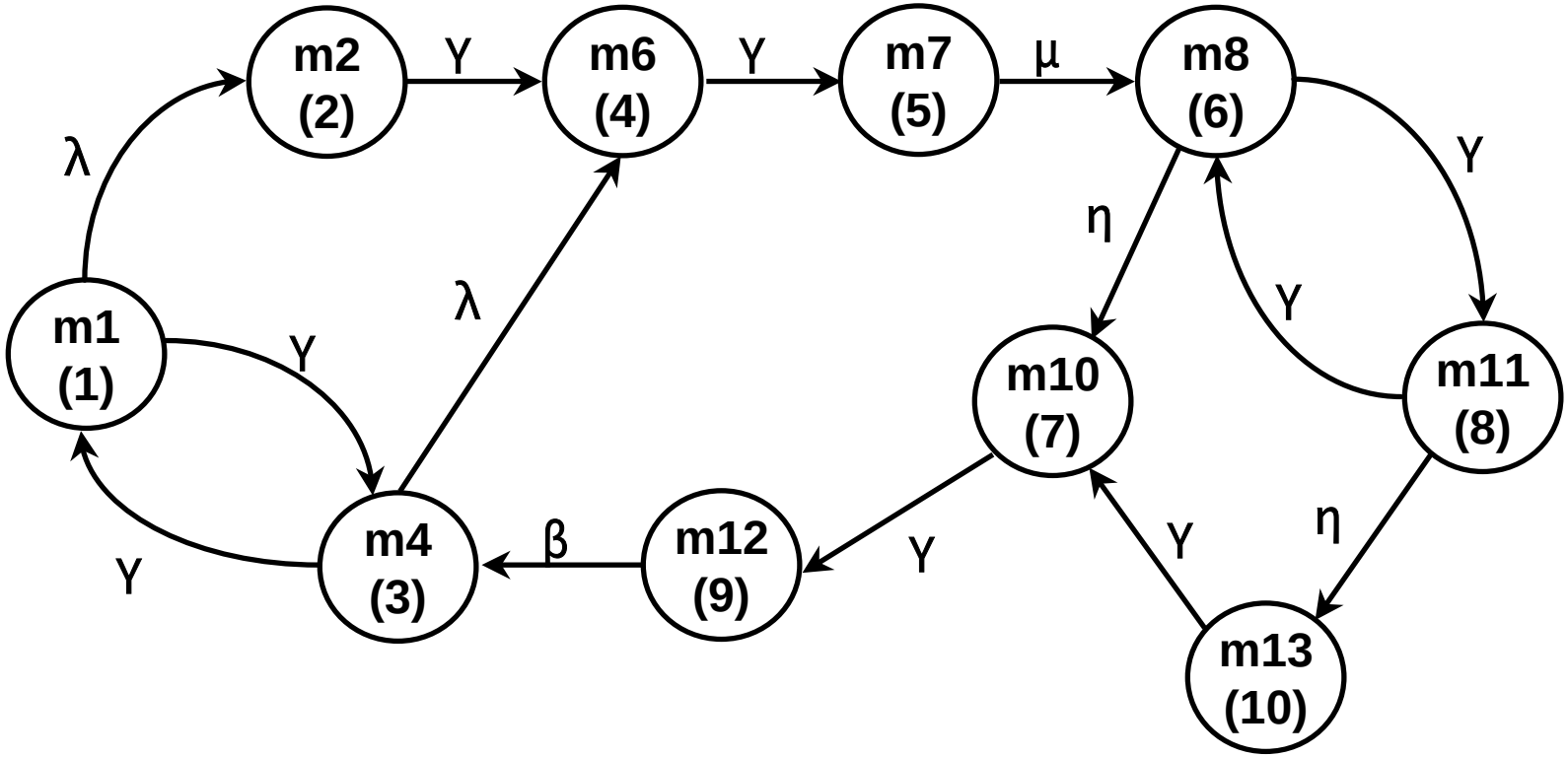
Reachability Graph



Reduced Reachability Graph



CTMC



Infinitesimal Generator Matrix

$$Q = \begin{bmatrix}
-(\lambda+\gamma) & \lambda & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\gamma & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 \\
\gamma & 0 & -(\lambda+\gamma) & \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\gamma & \gamma & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\mu & \mu & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -(\eta+\gamma) & \eta & \gamma & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\gamma & 0 & \gamma & 0 \\
0 & 0 & 0 & 0 & 0 & \gamma & 0 & -(\eta+\gamma) & 0 & \eta \\
0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & -\beta & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & -\gamma
\end{bmatrix}$$

Solution Methods

- Solve CTMC
  - $\pi = (\pi 1, \pi 2, \pi 3, \ldots, \pi 10)$: steady state probability vector of the CTMC
    - **π Q = 0**
    - Iterative method : Gauss-Seidel method, SOR method
  - **p(t)=(p1(t), p2(t), p3(t), . . . ,p10(t)) : transient probability vector**

$$\frac{dp(t)}{dt} = p(t)Q$$

    - 4th order Runge-Kutta ODE method (non stiff problem)
    - Uniformization, TR-BDF2 (stiff problem)

- Solve Markov reward models
  - r=(r1, r2, r3, . . . , r10) : vector of reward rate for each state
  - expected steady state reward $E[X] = \sum_{i \in S} r_i \pi_i$
  - expected transient(instantaneous) reward rate $E[X(t)] = \sum_{i \in S} r_i p_i(t)$
  - expected accumulated reward $E[Y(t)] = \sum_{i \in S} \int_0^t r_i p_i(\tau)\, d\tau$

계산 가능한 성능치(예)

- Mean probability of the server's being idle
- Mean probability of a specific client's being idle
- Mean response time at a client station
- Mean number of requests at the server
- Mean number of requests in the server's buffer
- Mean number of requests processed per second
  - Throughput of the server
  - Expected firing rate of the transition t4

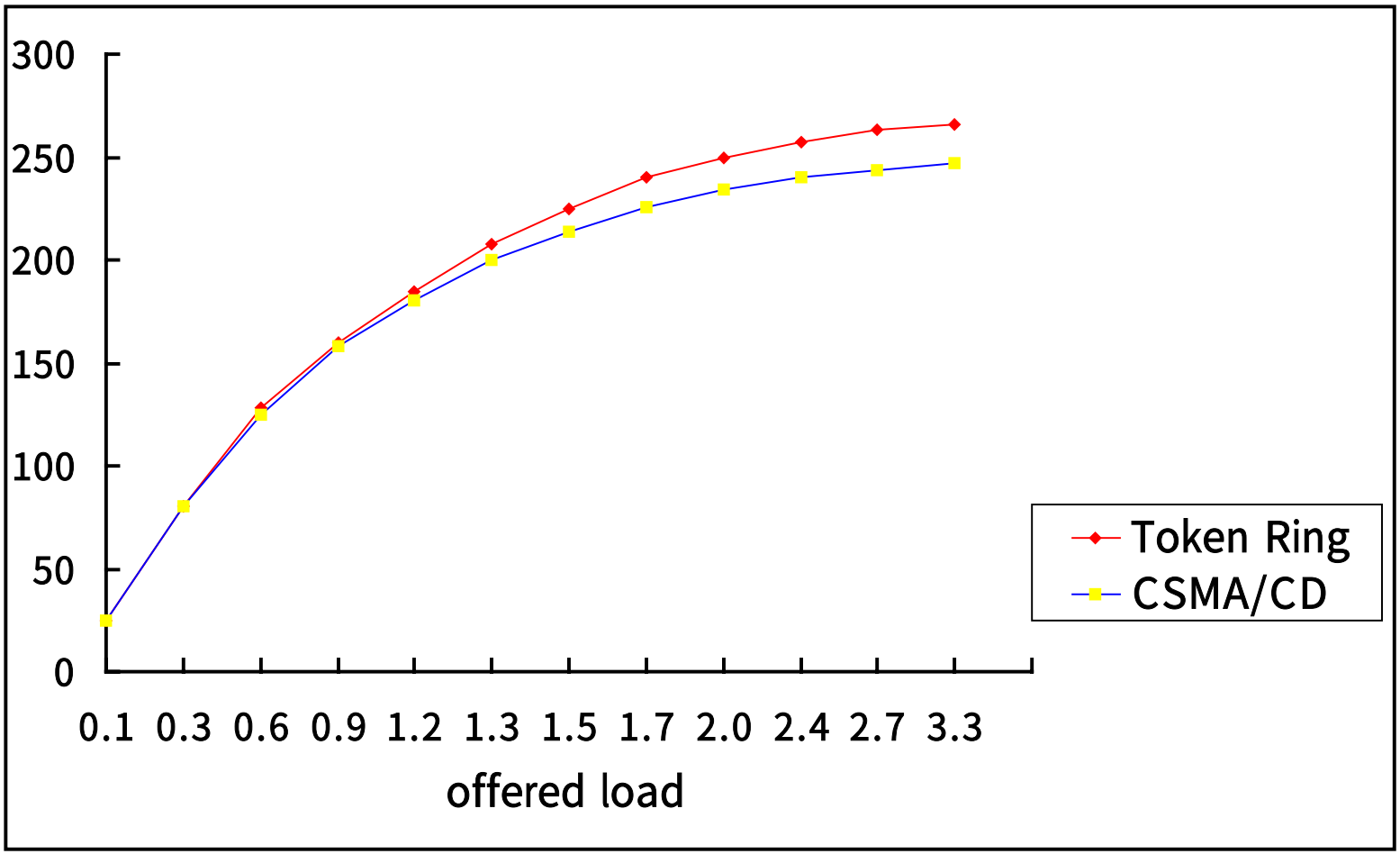
$$E[X] = \sum_{i \in S} r_i \pi_i$$

- if t4 is enabled, $r_i = \eta$
- if t4 is not enabled, $r_i = 0$

Throughput

- Example for 5 clients, 1 server system
- Mean length of request packet = 125 bytes/packet ($1/\mu$)
- Mean length of response packet = 4 Kbytes/packet ($1/\beta$)
- Mean polling time = 0.004ms ($1/\gamma$)
- Mean service time = 2ms ($1/\eta$)
- Request generation rate (λ) : variable
- Offered load, $\rho = \lambda(1/\mu + 1/\eta)$

Throughput


300
250
200
150
100
50
0
0.1 0.3 0.6 0.9 1.2 1.3 1.5 1.7 2.0 2.4 2.7 3.3
offered load
Token Ring
CSMA/CD

Introduction to Simulation

1. Designing Simulation Model
2. Programming Simulation
3. Analyzing Simulation Results

1. Designing Simulation Model

- A study of actions that mimics some reality.
- Probability based simulation
- Probability based simulations use Monte Carlo method.
- Steps of simulation
  - ① Designing simulation model
  - ② Programming simulation model
  - ③ Analyzing simulation results
- Simulations are complicated programs that will execute for a large number of iterations.
- Need to decide
  - Simulation method : physical simulation or Monte Carlo simulation
  - Model type
  - Characteristics of parameters
  - Model size
  - Simulation time (Number of iterations)

Types of simulation models

- Static model : States and characteristics of the model do not change over time.
  - Represents a system at a particular point in time. (Monte Carlo simulation)
- Dynamic model : States and characteristics of the model change over time.

- Deterministic model : Does not contain stochastic variables.
  - Known(deterministic) set of inputs that result in a unique set of outputs
  - Ex) Known input : arrivals of patients at the scheduled appointment time.
- Stochastic model : Contains stochastic variables.

- Discrete model : System variables change in a discontinuous way over time.
  - Ex) a bank
- Continuous model : System variables change continuously over time.
  - Ex) a dam, a worm object losing its heat to the air
- Combined model : Some variables are discrete and some are continuous.
  - Ex) a dam with water gates to outflow

Generating Random Numbers of Uniform Distribution

- In order to have a computer program to simulate a stochastic process, we need to generate numbers that represent values of random variables of events.
- Need algorithms to generate numbers randomly and independently based on some distribution.
- No generation algorithm can produce truly random numbers.

  : repeat of sequence of numbers

- Linear Congruential Method
  - *Zi* : random numbers a,c,m : constant Z(n+1) = (a*Zn+ c) modulo m
  - 예) Z1=7, a=9, c=11, m=50

    Z2= (9*7 + 11) mod 50= 24

    Z3= (9*24 +11) mod 50 = 27

- Mixed Congruential Method
  - The linear congruential method with c > 0

Multiplicative Congruential Method

- The linear congruential method with c = 0
- Periodic sequence appears sooner than the linear congruential method
- 예) Z1=13, a=9, m=16
  Z2 = (9*13) mod 16 = 5
  Z3 = (9* 5) mod 16 = 13
  Z4 = (9*13) mod 16 = 5
- UNIX's rand() uses this method with period 2^32 that returns successive pseudo-random numbers in the range from 0 to (2^15)-1.

Additive Congruential Method

- A new number is generated by the last generated number to the k-th previous number
- Zn = (Z(n-1) + Z(n-k)) mod m

Midsquare method

- Take the middle digits of the square of a number
- 예) a number 12 ---> the square 0144 ---> the random number 14

Generating Non-uniform Random Numbers

- Exponentially distributed random numbers
  - Y~EXP(λ) Z~UNIFORM(0,1) y= (-1/ λ)*ln (1-z)

$$F_Y(y) = P[Y \le y] = P[-\lambda^{-1}\ln(1-z) \le y]$$
$$= P[\ln(1-z) \ge -\lambda y]$$
$$= P[1-z \ge e^{-\lambda y}]$$
$$= P[z \le 1-e^{-\lambda y}]$$
$$= F_Z(1-e^{-\lambda y})$$

  - Since Z is uniformly distributed over (0,1), $F_Z(z)=z$, $0 <= z <= 1$. Thus :

$$F_Z(1-e^{-\lambda y}) = 1-e^{-\lambda y}$$

  - Therefore, Y is exponentially distributed with parameter λ.

- Random numbers of Poisson distribution
  - N~POI(λ , t) T~EXP(λ)
  - Count the number of exponentially distributed random variables needed to add up t.

- Random numbers of Geometric distribution
  - N~GEO(p) p:failure probability Z~UNIFORM(0,1)
  - Count the number of times that we generate uniformly distributed random numbers until we get a number greater than p.

2. Programming Simulations

- Basic structures of simulation programs
  1. Time-based simulation
  2. Event-based simulation

- Time-based simulation
  - Program control loop is associated with time.
  - Simple but inefficient.
  - Basic structure : Figure 1

- Event-based simulation
  - Execution of the main control loop represents a single event.
  - Need event queue to maintain the information to decide which event is next .
  - Complex but efficient and accurate.
  - Basic structure : Figure 2

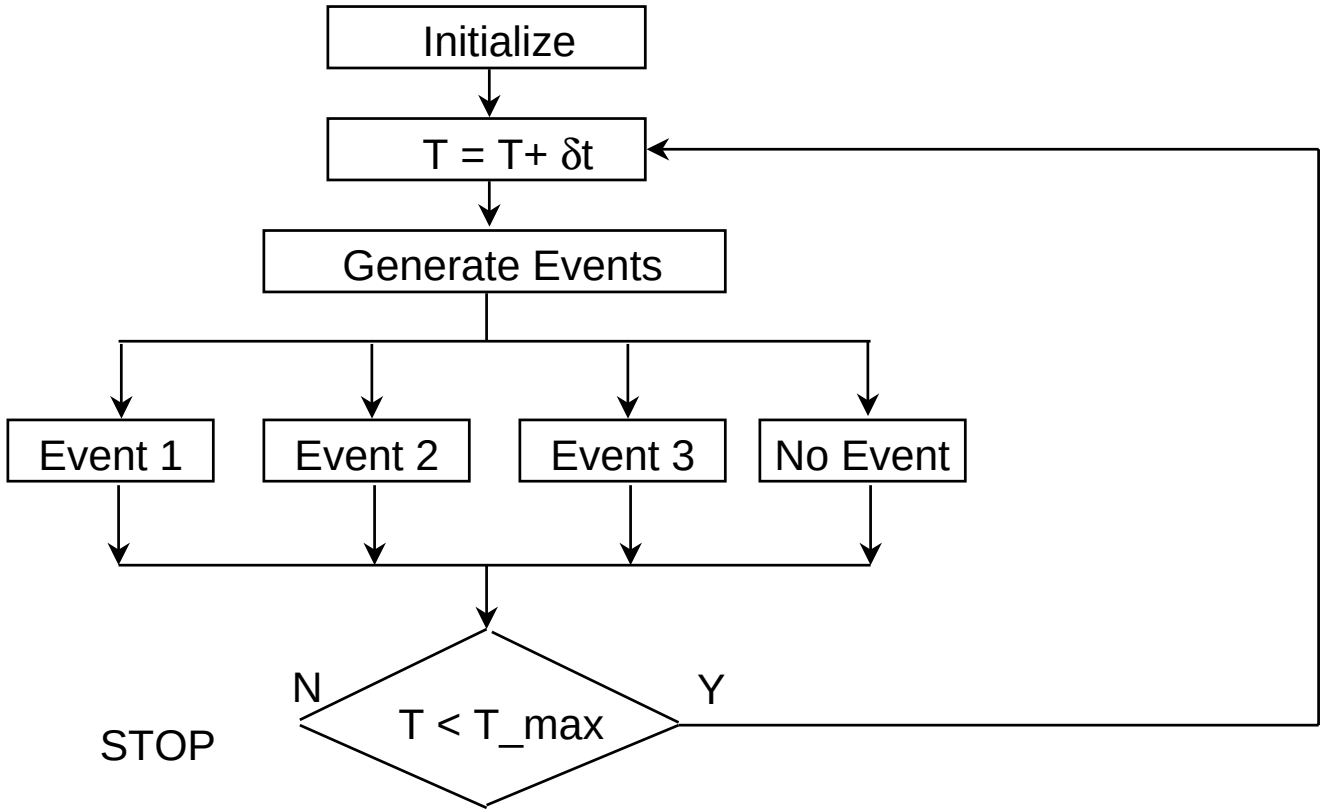


Figure 1 Control Flow of Time-Based Simulation

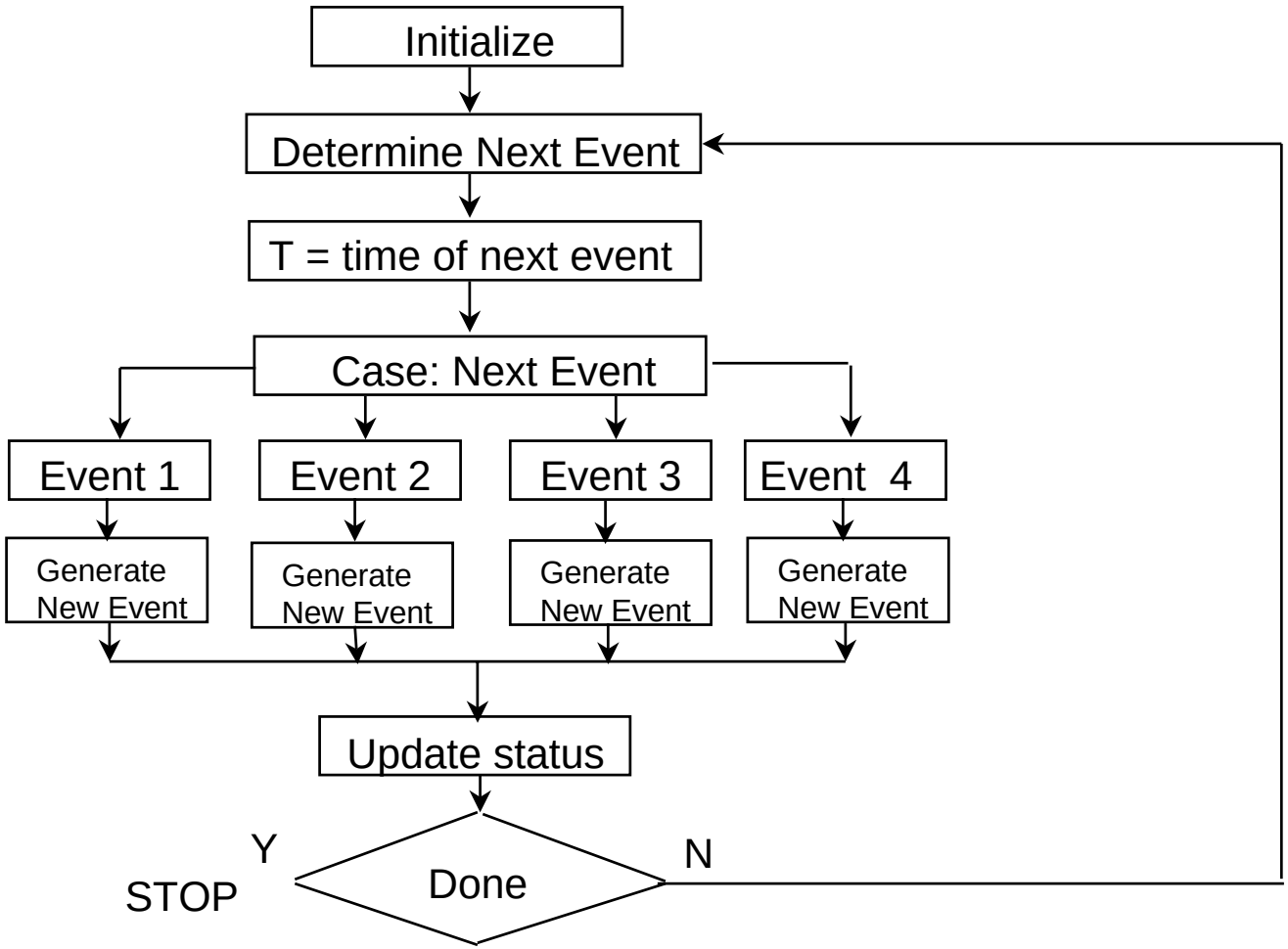


Figure 2 Control Flow of Event-Based Simulation

Simulation program of M/M/1 queuing system

```
main() {
    double Ta=200.0, Ts=100.0, te=200000.0, t1,t2,time;
    double expntl();
    int n;
    n=0; t1=0.0; t2=te;  time=0.0;
    while (time < te) {
            if (t1 < t2) {          /* arrival event */
                    time=t1;  n++;  t1=time+expntl(Ta);          /* t1:arrival time of next job */
                    if (n==1) t2=time+expntl(Ts);                /* t2:service time of this job */
            }
            else {                  /* service completion */
                    time=t2;  n--;      /* decrease the number of customers in the sy */
                    if (n > 0) t2=time+expntl(Ts);  else t2=te; /* let  next event be arrival */
            }
    }
}
double expntl(t)  double t; {
    double ranf();
    return( -t*log( ranf() ) );
}
```

Accumulating Statistics

- Since traces are voluminous, it is required to provide summary statistics about a simulation.
  - Adding program segments for the accumulation of statistics in a simulation program.
  - Calculating mean, standard deviation, pdf(histogram), accumulation of a specific statistics.
- Cannot get the additional statistics without re-running the simulation.
- Potential numerical problems

Simulation Tools

- SIMSCRIPT
  - The first version by RAND Corp. In the early 1960s.
  - The recent version is SIMSCRIPT II.5 by CACI Product Company(1994).
- MODSIM(Modern Simulator)
  - Developed by CACI based on object-oriented concepts.
  - Newer version is named SIMOBJECT.
- GASP IV
  - A collection of library routines written in FORTRAN.
- GPSS(General Purpose Simulation System)
  - IBM developed the first version in 1961 and the last version in 1970(GPSS V).
  - GPSS/H by Wolverine Software (‘77), GPSS/PC by Minuteman Software(‘84)
- SLAM(Simulation Language for Alternative Modeling)
  - Evolved from GASP.
- Other tools
  - SIMAN, EZSIM, Sim++, DEVSIM++, SIMNET, SIMFACTORY, COMNET, OPNET, BONES, …...

- Simulation languages
  - Special purpose simulation language : PAWS, SCERT II, RESQ
  - General purpose programming language : C++, C, FORTARN, Pascal
  - General purpose simulation language : GPSS, SIMSCRIPT II.5, Simula

- Simulation tools
  - GPSS (General Purpose Simulation System)
  - CPSim, POSES++, EZSIM, DEVSIM++ , Sim++, SHARPE, SLAM II
  - OPNET, Bones
  - COVERS
  - And many others……

3. Analyzing Simulation Results

- Problems in analyzing simulation results
  - How to determine the accuracy of the reported statistics?
  - How to determine the duration of the simulation run?
  - Startup transients : unusual pattern before the simulation reaches steady state
  - Repeated runs with same starting values for variables : identical events and statistics.
- Central limit theorem
  - The mean of a sample of *n* mutually independent random variables drawn from a population that has a mean of μ and a variance of σ^2, is approximately distributed as a with a mean of μ and σ^2.
  - If the simulation runs or batches of a simulation run are long enough, the output variables tend toward a normal distribution.
  - How long is enough is not easy to determine, it is application dependent.
- Sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Sample variance

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Confidence intervals
  - If the collection of n output values are independent and identically distributed with a normal distribution, then the actual mean μ would fall with probability p within an confidence interval $\pm\varepsilon$ (p : **confidence level, confidence coefficient**).

$$P[\ \bar{x} - \varepsilon < \mu < \bar{x} + \varepsilon\ ] = p$$

  - ε is given by the Student-t distribution normalized by the experimental variance s

$$\varepsilon = t_{(n-1,p)} \times \sqrt{\left(\frac{s^2}{n}\right)}$$

예) 950 ±30 with confidence level 0.95

45.5 % ±3 % with confidence level 0.92