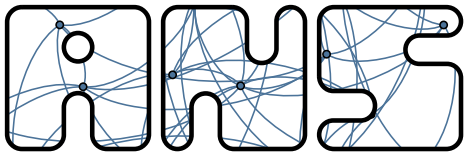


UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Solving & Using Markov Chains

Renato Lo Cigno

Simulation and Performance Evaluation 2017-18

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We have seen many (. . . well a few) techniques to derive a mathematical model
- Markov Chains are one of these, but how can we use them to derive performance and prediction?
- An MC can always be simulated . . . you will actually do that in the second assignment, even if the MC is somehow "hidden" within the code

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We have seen many (. . . well a few) techniques to derive a mathematical model
- Markov Chains are one of these, but how can we use them to derive performance and prediction?
- An MC can always be simulated . . . you will actually do that in the second assignment, even if the MC is somehow "hidden" within the code
- Some (many?) MC can be solved analytically
- Properties (or metrics, or rewards) associated to states or transitions provide the means for PE & predictions

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- There are different solutions of MCs, and DT or CT change slightly the methodology

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- There are different solutions of MCs, and DT or CT change slightly the methodology
- Steady State solution
  - Based on the regime distribution probability over states
  - Independent from the initial state
  - Gives insight on the "average" performance of the system

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- There are different solutions of MCs, and DT or CT change slightly the methodology
- Steady State solution
  - Based on the regime distribution probability over states
  - Independent from the initial state
  - Gives insight on the "average" performance of the system
- Transient solution
  - Function of the initial state
  - Describes the short-term temporal evolution of the system

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- There are different solutions of MCs, and DT or CT change slightly the methodology
- Steady State solution
  - Based on the regime distribution probability over states
  - Independent from the initial state
  - Gives insight on the "average" performance of the system
- Transient solution
  - Function of the initial state
  - Describes the short-term temporal evolution of the system
- We concentrate on steady state

Solving a DTMC

- We know that the evolution of a Markov Chain depends only on the state . . . and we assume a time-homogeneous DTMC to make things simpler
- States are numerable, so without loss of generality we can set $S = \{0, 1, 2, 3, 4, \ldots\}$
- p_{jk} denotes the transition probability from state j to state k

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We know that the evolution of a Markov Chain depends only on the state . . . and we assume a time-homogeneous DTMC to make things simpler
- States are numerable, so without loss of generality we can set $S = \{0, 1, 2, 3, 4, \ldots\}$
- p_{jk} denotes the transition probability from state j to state k
- The matrix

$$P = [p_{ij}] = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdot & \cdot \\ p_{10} & p_{11} & p_{12} & \cdot & \cdot \\ p_{20} & p_{21} & p_{22} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

completely characterized a DTMC

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- P is a *stochastic matrix*, i.e., it has the following properties

$$0 \leq p_{ij} \leq 1, \quad \forall i, j \in S$$

$$\sum_{j \in S} p_{ij} = 1, \quad \forall i \in S$$

- P elements are all non-negative
- P rows **must** sum to 1 for the theorem of total probability (i.e., the sum of the probabilities of disjoint events covering S must be 1)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- P is a *stochastic matrix*, i.e., it has the following properties

$$0 \leq p_{ij} \leq 1, \quad \forall i, j \in S$$

$$\sum_{j \in S} p_{ij} = 1, \quad \forall i \in S$$

- P elements are all non-negative
- P rows **must** sum to 1 for the theorem of total probability (i.e., the sum of the probabilities of disjoint events covering S must be 1)

- **Representing a DTMC with P or with the state diagram is exactly the same**

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Let $\mathbf{p}(n) = [p_0(n), p_1(n), \ldots, p_j(n), \ldots]$ be the vector of the probability of being in a given state at step n
- Clearly $\sum_{i \in S} p_i(n) = 1, \quad \forall n$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Let $\mathbf{p}(n) = [p_0(n), p_1(n), \ldots, p_j(n), \ldots]$ be the vector of the probability of being in a given state at step n
- Clearly $\sum_{i \in S} p_i(n) = 1, \quad \forall n$
- It is immediate to see that

$$\mathbf{p}(n+1) = \mathbf{p}(n)P$$

State Probabilities

- Let $\mathbf{p}(n) = [p_0(n), p_1(n), \ldots, p_j(n), \ldots]$ be the vector of the probability of being in a given state at step n
- Clearly $\sum_{i \in S} p_i(n) = 1, \quad \forall n$
- It is immediate to see that

$$\mathbf{p}(n+1) = \mathbf{p}(n)P$$

- If we have an initial state distribution (e.g., $\mathbf{p}(0) = [1, 0, 0, 0, \ldots]$) with a simple recursion we have

$$\mathbf{p}(1) = \mathbf{p}(0)P; \;\; \mathbf{p}(2) = \mathbf{p}(1)P = \mathbf{p}(0)P^2; \;\; \ldots; \;\; \mathbf{p}(n) = \mathbf{p}(0)P^n$$

Chapman-Kolmogorov equations

- Another way to see the evolution of a DTMC is computing the transition probabilities in n steps $\forall n$
- This imply computing the sum of the probabilities of all possible paths to go from state i to state j in exactly n-steps

Chapman-Kolmogorov equations

- Another way to see the evolution of a DTMC is computing the transition probabilities in n steps $\forall n$
- This imply computing the sum of the probabilities of all possible paths to go from state i to state j in exactly n-steps
- For $n = 1$ this is trivially p_{ij} entry of the transition matrix P

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Another way to see the evolution of a DTMC is computing the transition probabilities in n steps $\forall n$
- This imply computing the sum of the probabilities of all possible paths to go from state i to state j in exactly n-steps
- For $n = 1$ this is trivially p_{ij} entry of the transition matrix P
- Recall that for a time-homogeneous DTMC by definition

$$p_{ij}(n) = \mathbf{P}[X_{m+n} = j | X_m = i], \quad \forall m$$

so we can drop the dependence on m

$$p_{ij}(n) = \mathbf{P}[X_n = j | X_0 = i]$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The equation above tells us that we have to compute all the conditional probabilities of going from state i to state k in h steps times the probability of going from state k to state j in $n - h$ stpes

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The equation above tells us that we have to compute all the conditional probabilities of going from state i to state k in h steps times the probability of going from state k to state j in $n - h$ stpes
- Formally

$$p_{ij}(n) = \sum_{h=1}^{n} \sum_{k \in S} p_{ik}(h) p_{kj}(n-h)$$

which are the Chapman-Kolmogorov equations that can be rewritten in the simple matrix form of

$$P(n) = P \cdot P(n-1) = P^n$$

in case of homogeneous DTMC

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We can ask the question if it is possible (and meaningful) to compute

$$\mathbf{v} = [v_0, v_1, \ldots v_i, \ldots]$$

where

$$v_i = \lim_{n \to \infty} p_i(n)$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We can ask the question if it is possible (and meaningful) to compute

$$\mathbf{v} = [v_0, v_1, \ldots v_i, \ldots]$$

where

$$v_i = \lim_{n \to \infty} p_i(n)$$

- As $\mathbf{p}(n) = \mathbf{p}(0)P^n$, it equivalent to ask if $\lim_{n\to\infty} P^n$ exists and is meaningful

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- We can ask the question if it is possible (and meaningful) to compute

$$\mathbf{v} = [v_0, v_1, \ldots v_i, \ldots]$$

where

$$v_i = \lim_{n \to \infty} p_i(n)$$

- As $\mathbf{p}(n) = \mathbf{p}(0)P^n$, it equivalent to ask if $\lim_{n\to\infty} P^n$ exists and is meaningful
- If these limits exists and are meaningful, as P is a stochastic matrix and $\mathbf{v}$ is a stochastic vector $\mathbf{v}$ is the left eigenvector of P associated to the eigenvalue $\lambda = 1$ and can be found as

$$\mathbf{v} = \mathbf{v}P$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Every vector $\mathbf{v}$ that satisfies

$$\mathbf{v} = \mathbf{v}P; \quad \sum_{i \in S} v_i = 1$$

is called a *stationary distribution* (or probability) of the DTMC

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Every vector $\mathbf{v}$ that satisfies

$$\mathbf{v} = \mathbf{v}P; \quad \sum_{i \in S} v_i = 1$$

  is called a *stationary distribution* (or probability) of the DTMC

- If $\mathbf{v}$ exists, it is unique and independent from the initial state $\mathbf{p}(0)$ of the DCMC, then it is called the *steady-state* of the DTMC

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Every vector $\mathbf{v}$ that satisfies

$$\mathbf{v} = \mathbf{v}P; \quad \sum_{i \in S} v_i = 1$$

  is called a *stationary distribution* (or probability) of the DTMC

- If $\mathbf{v}$ exists, it is unique and independent from the initial state $\mathbf{p}(0)$ of the DCMC, then it is called the *steady-state* of the DTMC

- Question: Under which conditions the steady-state of a DTMC exists?

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

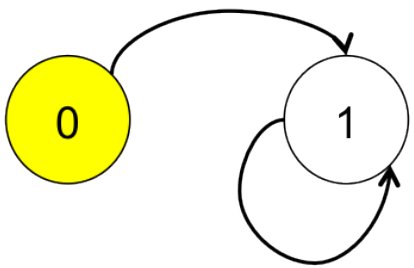
Definition: Transient State

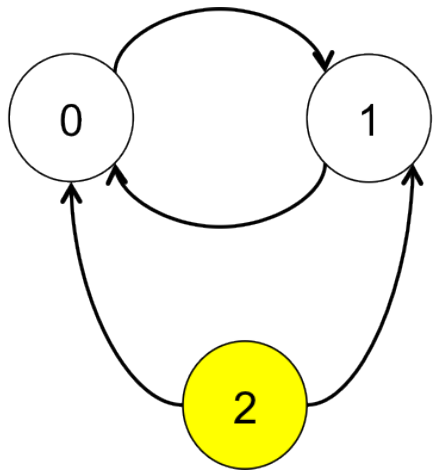
- A state i is said to be transient if there is a positive probability that the process will never return to i after leaving it
- Formally this is equivalent to state that

$$\lim_{n\to\infty} p_{ji}(n) = 0; \quad \forall j \in S$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Transient States (yellow)





UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent State

- A non-transient state is said recurrent
- A state is recurrent if the probability of visiting i after leaving it for $n \to \infty$ is 1

$$\sum_{n=1}^{\infty} p_{ii}(n) = \infty$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent State

- Let $f_{ij}(n)$ be the conditional probability that the first visit to j after leaving i occurs in exactly n steps
- Then the probability of ever visiting state j (sooner or later) starting from state i is

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}(n)$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent State

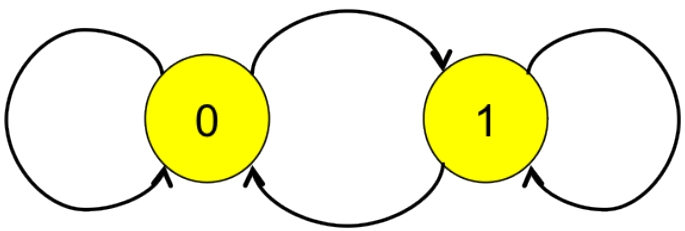
- Let $f_{ij}(n)$ be the conditional probability that the first visit to j after leaving i occurs in exactly n steps
- Then the probability of ever visiting state j (sooner or later) starting from state i is

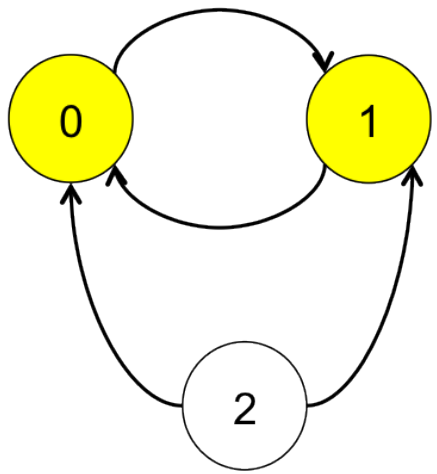
$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}(n)$$

- A state is recurrent if $f_{ii} = 1$; if $f_{ii} \leq 1$ it is transient

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Recurrent States (yellow)





UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent Positive State

- For a recurrent state i it is interesting to know the distribution of the recurrence time, i.e., after how many steps the DTMC returns to i after leaving it

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent Positive State

- For a recurrent state i it is interesting to know the distribution of the recurrence time, i.e., after how many steps the DTMC returns to i after leaving it
- We define the *mean recurrence time* of state i

$$\mu_i = \sum_{n=1}^{\infty} n f_{ii}(n)$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Recurrent Positive State

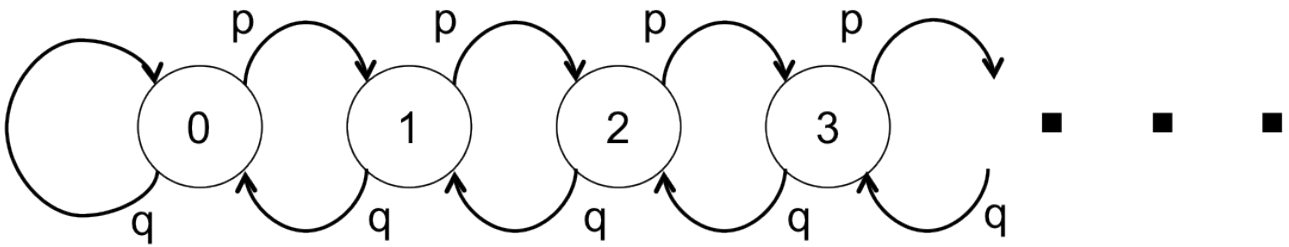
- For a recurrent state i it is interesting to know the distribution of the recurrence time, i.e., after how many steps the DTMC returns to i after leaving it
- We define the *mean recurrence time* of state i

$$\mu_i = \sum_{n=1}^{\infty} n f_{ii}(n)$$

- A state is said *recurrent positive* (non-null) if $\mu_i \leq \infty$
- A state is said *recurrent null* if $\mu_i = \infty$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Recurrent Null/Positive States



- If $p < q$ all states are recurrent positive
- If $p \geq q$ all states are recurrent null

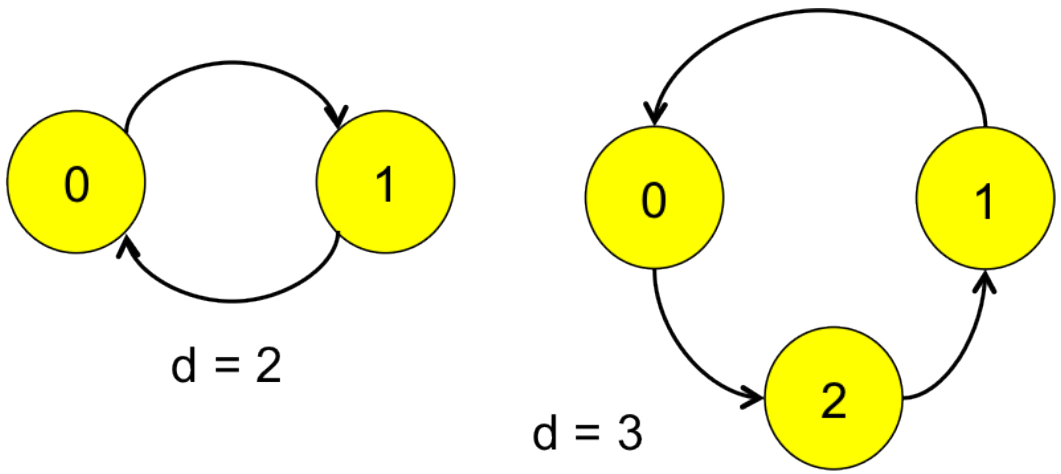
UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Periodic State

- Let d_i be the greatest common divisor of the set of positive integers n such that $p_{ii}(n) > 0$
- A state is said *periodic* if $d_i > 1$; the value d_i is called the period
- A state is said *aperiodic* if $d_i = 1$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Periodic States (yellow)



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Definition: Absorbing and Communicating States

- A state i is said *absorbing* if $p_{ii} = 1$
- Once the DTMC enters i it will never leave it
- This notion can be extended to a set of states

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

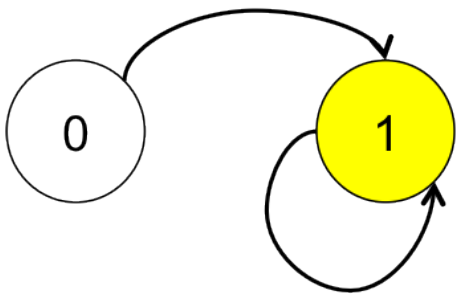
Definition: Absorbing and Communicating States

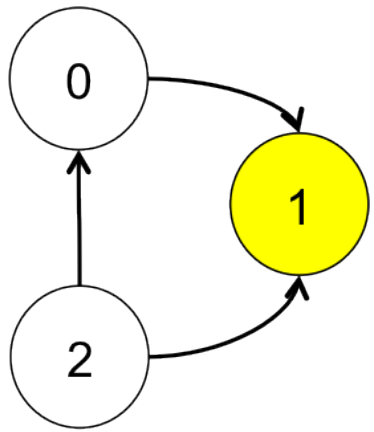
- A state i is said *absorbing* if $p_{ii} = 1$
- Once the DTMC enters i it will never leave it
- This notion can be extended to a set of states

- Given two states i and j they are said *communicating* if directed paths exist from i to j and viceversa $p_{ij}(n) > 0$ for some n and $p_{ji}(m) > 0$ for some m

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

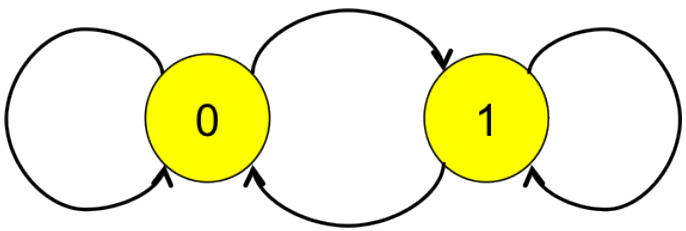
Absorbing States (yellow)



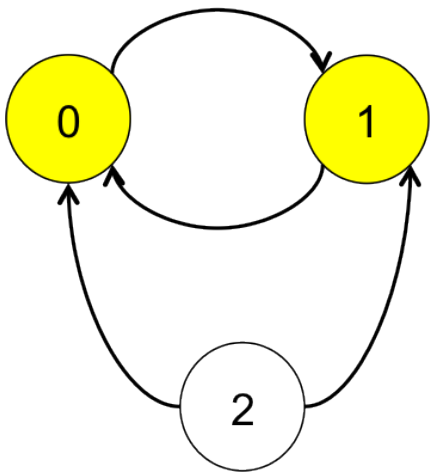


UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

Communicating States
(yellow)

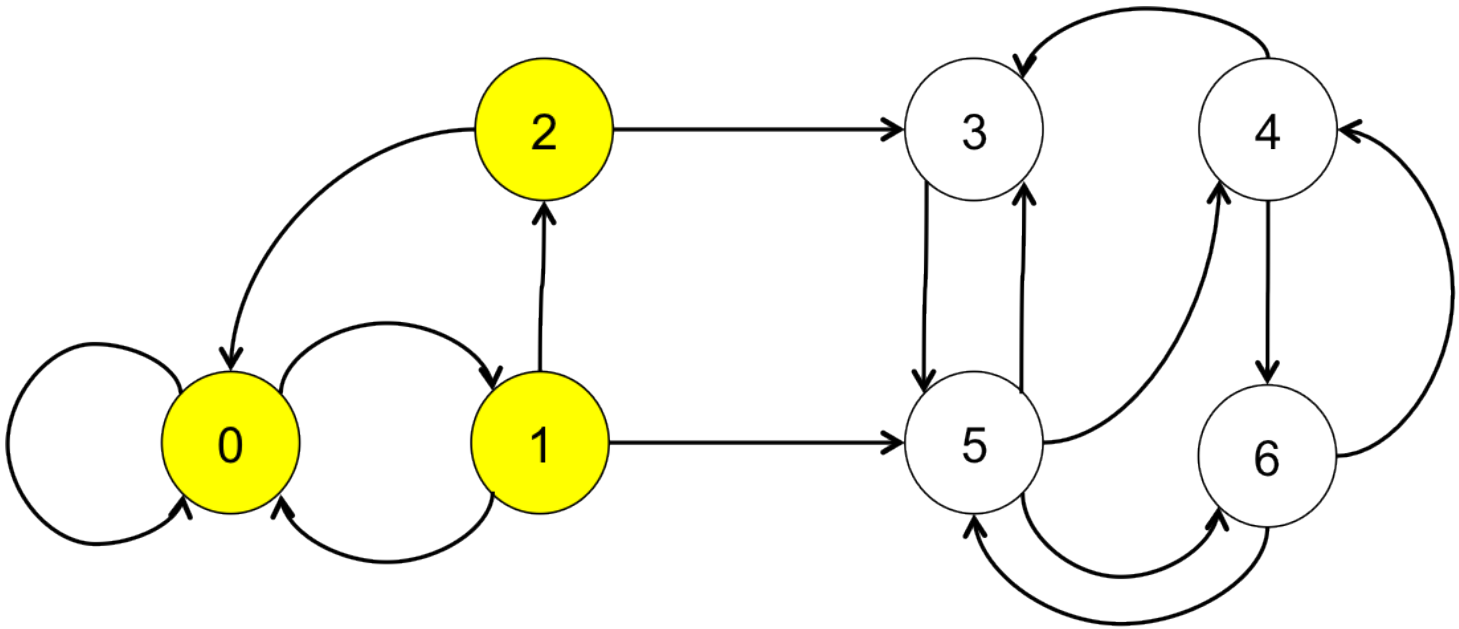


Non Communicating States
(yellow, 0 and 1 do not communicate with 2)



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

DTMC with Transient States (yellow) and a set of absorbing states (white) that do not communicate with the Transient ones



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

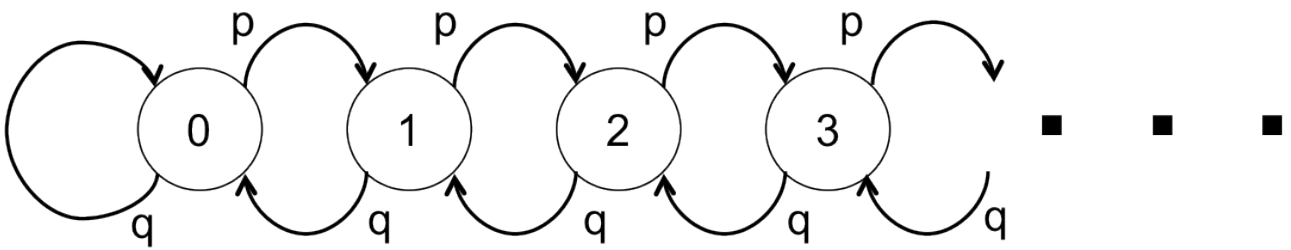
- A MC (not only DT) is said **Irreducible** if every state i is reachable from any other state j in finite time: $\forall i, j \in S$ there exists $n \geq 1$ such that $p_{ij}(n) > 0$
- An irreducible MC does not have Transient or recurrent-null states, i.e., they are all recurrent positive states
- All states in an irreducible MC are of the same type: Periodic or Aperiodic

Any Irreducible Aperiodic Markov Chain admits a Steady-State that can be computed (for DTMCs) as

$$\mathbf{v} = \mathbf{v}P$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- If $|S|$ is infinite, then the steady state can be found only if P has some special structure that allows a recursive solution
- Example: DT Birth-Death Process with $p < q$



$$P = \begin{bmatrix} q & p & 0 & 0 & 0 & \cdot & \cdot & \cdot \\ q & 0 & p & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & q & 0 & p & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & q & 0 & p & \cdot & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- To solve the system we simply have to solve this system of recursive equations

$$\begin{cases} pv_0 &= q(v_0 + v_1) \\ (p+q)v_i &= pv_{i-1} + qv_{i+1} \quad \forall i > 0 \\ \sum_{i=0}^{\infty} v_i &= 1 \end{cases}$$

- Whose solution yields the well known geometric distribution of customers in a queue:

$$v_0 = \left(1 - \frac{p}{q}\right); \quad v_i = \left(1 - \frac{p}{q}\right)\left(\frac{p}{q}\right)^i \quad \forall i > 0$$

- The DT Birth-Death Process models any (single server, single customer class) DT queueing system given that p is known and $q = (1 - p)$ is a reasonable assumption

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Consider a simple processor with two cores and a L1 cache memory
- If processes running on different cores need to access the cache there is a conflict and one must wait slowing the processing

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The state of the system is simply
  $S = \{I, C_1, C_2, W\} = \{0, 1, 2, 3\}$
  Idle: no core is accessing the cache; C_1 (C_2) core 1 (or 2) is accessing alone; or one is accessing and the other is Waiting
- Assume the probability of accessing the cache are p_1 and p_2 respectively in any time slot and the time to retrieve the content of the cash is exactly one slot time, while retrieving the content the core is blocked and cannot generate other requests. Requests are independent. Then the model is

$$P_W = \begin{bmatrix} 1-(p_1+p_2) & p_1 - 0.5p_1p_2 & p_2 - 0.5p_1p_2 & p_1p_2 \\ 1-p_2 & 0 & p_2 & 0 \\ 1-p_1 & p_1 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Is the model represented by the P matrix in the slide before correct?

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

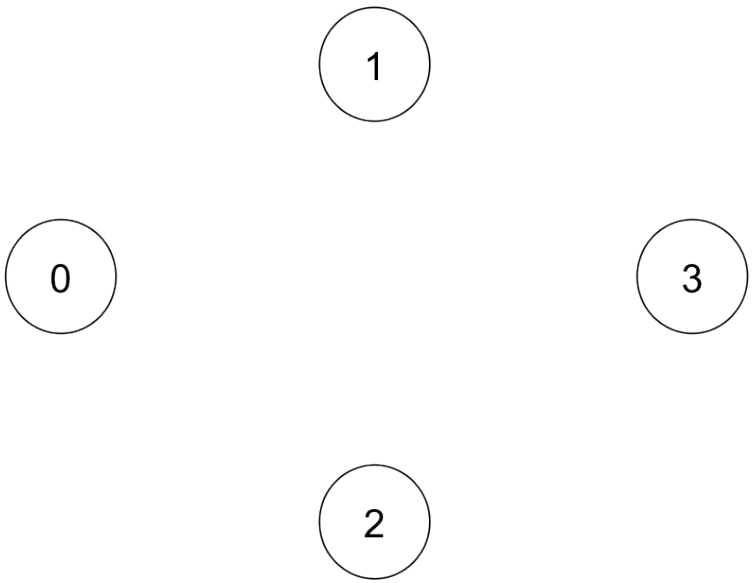
- Is the model represented by the P matrix in the slide before correct?
- No, it cannot be: simply observing that for $p_1 + p_2 > 1$ (perfectly legitimate) then $p_{00} < 0$
- The correct model is obviously

$$P_C = \begin{bmatrix} 1-(p_1+p_2-p_1p_2) & p_1(1-p_2) & p_2(1-p_1) & p_1p_2 \\ 1-p_2 & 0 & p_2 & 0 \\ 1-p_1 & p_1 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \end{bmatrix}$$

- Let's see why

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

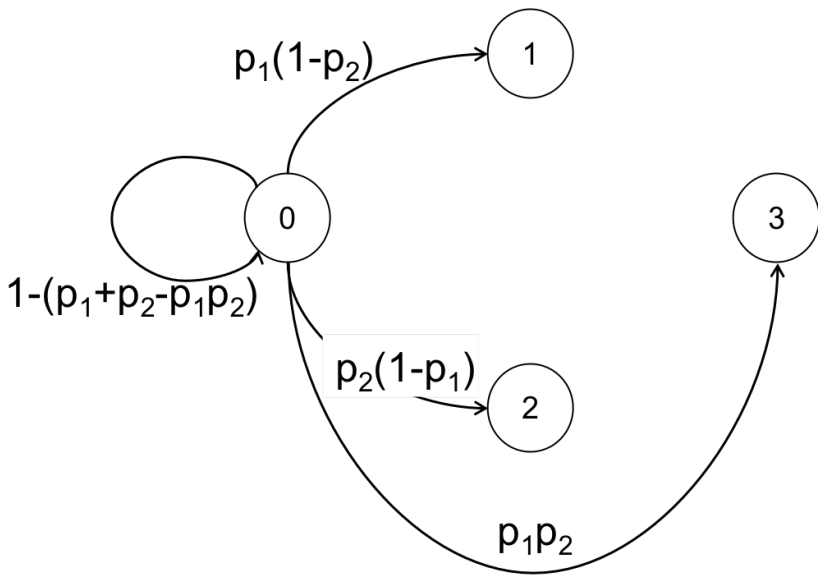
The state space



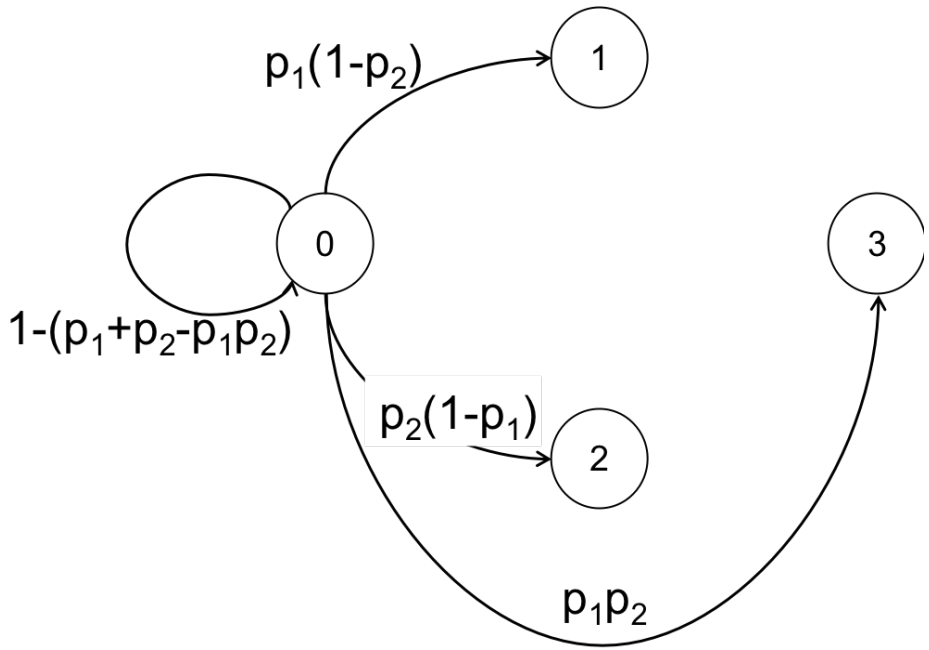
UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

If no cores are accessing the cache, then we have four possibilities

- C1 requests access and C2 does not: $p_{01} = p_1(1 - p_2)$
- C2 requests access and C1 does not: $p_{02} = p_2(1 - p_1)$
- C1 and C2 request together: $p_{03} = p_1 p_2$
- None of them request access: $p_{00} = 1 - (p_1 + p_2 - p_1 p_2)$



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

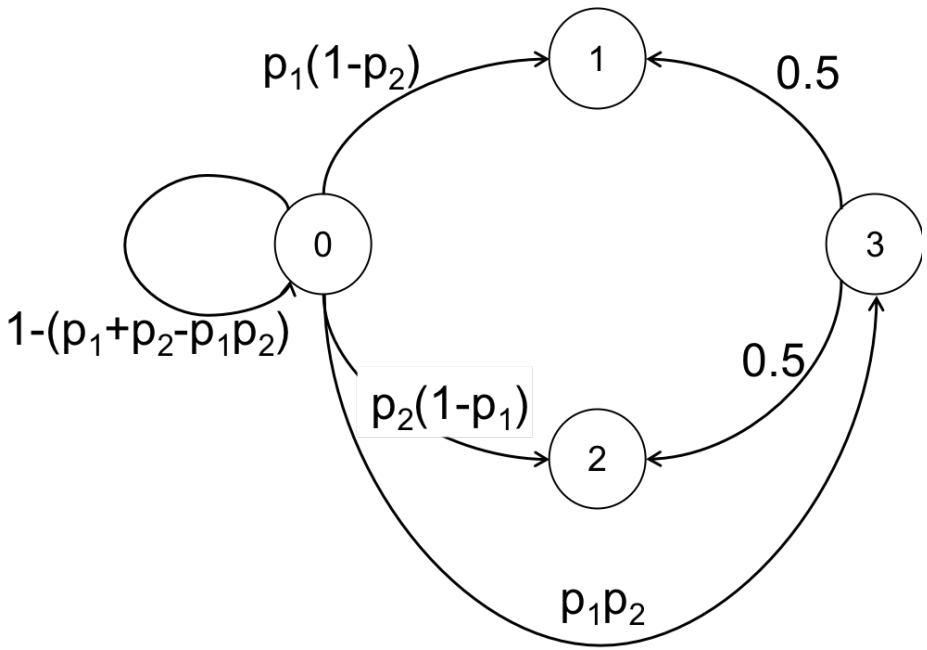


Here we already see the mistake in the model P_W (W stands for wrong, not wait ...); in P_W the probability assignment was not done considering correctly the independence of access from the two cores, and then p_{00} was "twisted" to force the sum of the transition probabilities to 1

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

After a block, either C1 or C2 are served first, with no reason to assign different probabilities to this events

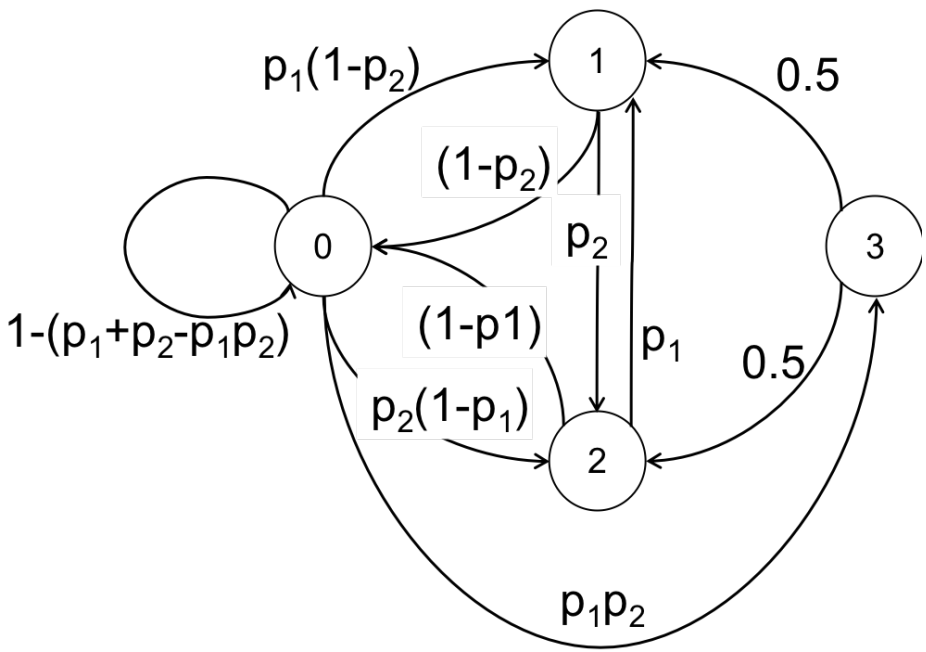
- $p_{31} = p_{32} = 0.5$



UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

And finally, after one core accessed the cache, either the other does or the system goes back to idle

- $p_{12} = p_2$; $p_{10} = (1 - p_2)$
- $p_{21} = p_1$; $p_{20} = (1 - p_1)$



Homework (not exam assignment!)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Solve the model
- Extend the model to 4 cores and content retrieve time uniformly distributed between 1 and 4 slots and solve it (if it is too complex to solve it in close form, program the solution as a function of $p_1 \cdots p_4$)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The real performance of the system can be normally derived from the state distribution (sometimes from transitions, but we do not consider this case for the time being
- We can associate **rewards** r_i to any state that measure it performance
- The performance of the system is associated to the average reward r

$$r = \sum_{i=0}^{\infty} r_i \cdot v_i$$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The real performance of the system can be normally derived from the state distribution (sometimes from transitions, but we do not consider this case for the time being
- We can associate **rewards** r_i to any state that measure it performance
- The performance of the system is associated to the average reward r

$$r = \sum_{i=0}^{\infty} r_i \cdot v_i$$

- If we are interested in the transient reward until step K we can compute

$$r(K) = \sum_{k=0}^{K} \sum_{i=0}^{\infty} r_i \cdot p_i(k)$$

Homework (not exam assignment!)

- Back to the cache memory model
- The performance of the system is given by its efficiency, so we can assume the following reward distribution:
  $r_0 = 1, r_1 = r_2 = 0.5; r_3 = 0$

Homework (not exam assignment!)

- Back to the cache memory model
- The performance of the system is given by its efficiency, so we can assume the following reward distribution:
  $r_0 = 1, r_1 = r_2 = 0.5; r_3 = 0$
- Compute the "surface" (p_1, p_2) that guarantees that $r > r_t$, where r_t is the target efficiency of your system
- This result tells you what are the characteristics of the workload that your 2-core processor can accept

Homework (not exam assignment!)

- Back to the cache memory model
- The performance of the system is given by its efficiency, so we can assume the following reward distribution:
  $r_0 = 1, r_1 = r_2 = 0.5; r_3 = 0$
- Compute the "surface" (p_1, p_2) that guarantees that $r > r_t$, where r_t is the target efficiency of your system
- This result tells you what are the characteristics of the workload that your 2-core processor can accept
- Extend this result to the 4 cores case
- Make a comparison between a 4 core processor and two 2-cores one with the same processing power and cache capacity

What if the time is Continuous

- Some systems cannot be modeled in discrete time . . .
  - When "human time" is involved
  - When the evolution of the system is intrinsically analogic
- . . . but we know there are CTMC
- Classification of CTMC states is similar to DTMC, but Periodic states do not exist
- The condition for steady state existence is similar to DTMCs (we do not make the whole analysis again)

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- Recall that all transitions in a CTMS are exponentially distributed (implied by the fact that dwell times must be exponentially distributed)
- A CTMC is fully described by a matrix

$$Q = [q_{ij}] = \begin{bmatrix} q_0 & q_{01} & q_{02} & \cdot & \cdot \\ q_{10} & q_1 & q_{12} & \cdot & \cdot \\ q_{20} & q_{21} & q_2 & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

called the **infinitesimal generator**

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- q_{ij} are the transition rates from stat i to state j
- $q_i = -\sum_{j=0, j\neq i}^{\infty} q_{ij}$
- Neither q_{ij}, nor q_i are probabilities, but the relation above stems for a simple conservation law "on average whatever goes in must come out"
- State probabilities are normally called π and not v
  $\ldots \pi(t) = [\pi_0(t), \pi_1(t), \pi_2(t), \ldots]$

UNIVERSITY OF TRENTO
Department of Information Engineering and Computer Science

- The steady state of a CTMC exists under the same conditions (with the due changes!) of a DTMC
- The Chapman-Kolmogorov equations can be found first writing time-dependent probabilities and then taking the limit for δt going to zero, obtaining differential equations
- Finally, solving these equations we find that the steady-state state probability vector π as solution of the linear system

$$\pi Q = 0; \quad \sum_{i=0}^{\infty} \pi_i = 1$$