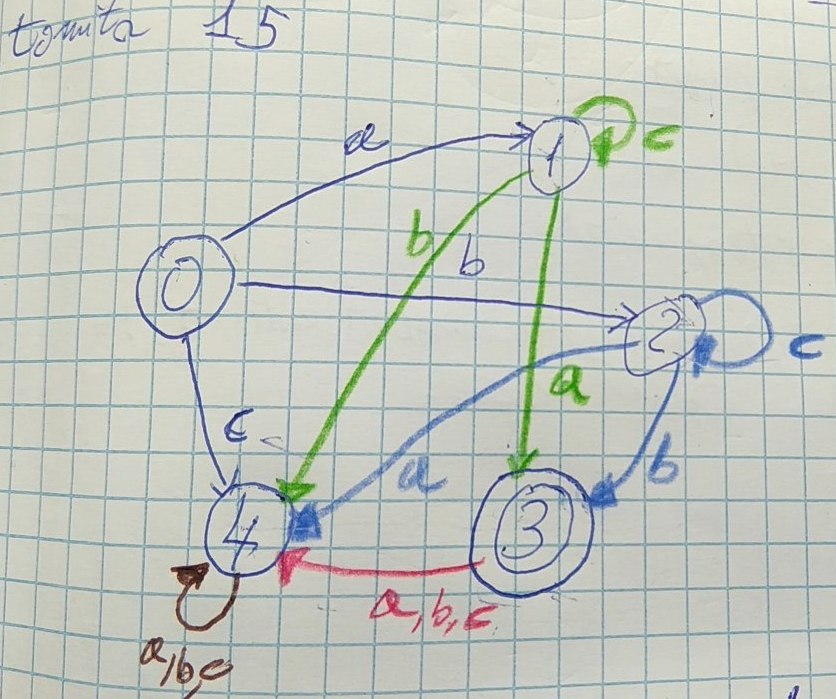
**Esempio faccenda stati accettanti/rigettanti**

****

Innanzitutto confrontando il tomita15 con sé stesso ritorna 1, bene.

Fissando il Tomita15 originale come reference dfa, (che ha solo lo stato 3 come accettante), faccio variare lo/gli stati accettanti del Tomita15 e l’automa che ottengo lo uso come subject. Riporto accanto il valore di similarità strutturale.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ||| \*\*\*\*\*\*\*\*\*\*\***

**Stati accettanti ||| Similarità**

**Tomita15 modified ||| strutturale**

0 0.000144239

1 0.290008

2 0.290008

3 (same as tomita15) 1

4 0.257799

none 0.378851

0,1 0.000101101

0,2 0.000101101

0,3 0.000405275

0,4 3.56063e-05

1,2 0.145027

1,3 0.340992

1,4 0.17786

2,3 0.340992

2,4 0.17786

3,4 0.270101

0,1,2 4.13084e-05

0,1,3 0.000163523

0,1,4 1.15549e-05

1,2,3 0.150086

1,2,4 0.100003

2,3,4 0.170869

0,1,2,3 6.16027e-05

0,1,2,4 1.85185e-06

1,2,3,4 0.100004

0,1,3,4 6.20058e-06

0,2,3,4 6.20058e-06

0,1,2,3,4 0

Che le coppie (1,4);(2,4) riportino lo stesso punteggio di similarità è un buon segnale perché i nodi 1 e 2 sono strutturalmente identici (stessi nodi sorgente che portano ad essi, stessi nodi terminazione a cui arrivano i loro archi, stessa combinazione di archi: uno che torna in sé stesso, uno che va nello stato 3, uno che va nello stato 4).

**Quello che succede con questa implementazione è che viene fatto un controllo nella funzione Iterate() e se nel grosso ciclo su tutte le coppie dei nodi dei grafi trova una coppia con "colore"/label diversa, quello che fa è semplicemente andare avanti nel for, infatti:**

for(int i=0; i<graph\_a\_n; i++)

for(int j=0; j<graph\_b\_n; j++)

{

if(graph\_a->Label(i)!=graph\_b->Label(j))

continue;

...

...

}

Non so se è quello che vogliamo, potremmo ripensare la faccenda prevedendo un certo coefficiente di penalità per il confronto tra acc/rej anziché il taglio netto.

Proviamo a ragionare con il nostro scenario:

mettiamo che Alice per andare a lavoro si dirige verso casa di Bob, e Bob per andare al lavoro si dirige verso casa di Alice e si mettono d'accordo per fare esattamente lo stesso percorso in maniera speculare, tra andata e ritorno.

Possiamo dire che Alice e Bob fanno la stessa strada?

O il differenziarsi dei tragitti tra andata e ritorno deve farci considerare diversi i percorsi? E se è così, poichè sono esattamente speculari, allora si somigliano al 50%?

**Altri esempi di calcolo di punteggio di similarità SENZA CONSIDERARE IL COLORE**

**Dal paper:**

The termination condition is for some chosen precision .

**Confronto 0**

Confronto il Tomita15 con un suo duplicato in cui dallo Stato 4, con la lettera b, andiamo allo stato 2.

Quello che vediamo è che incrementando il valore del fattore di terminazione epsilon, il valore di similarità diminuisce di pochissimo, segnalando di fatto la convergenza dell’algoritmo:

**Epsilon ||| Similarità**

0.01 ﷒ 0.443864

0.001 0.418553

0.0001 0.415896

… …

0.000000001 ﷒ 0.415584

… …

Aver cambiato una sola transizione ha più che dimezzato il punteggio di similarità. Ce lo aspettavamo? Forse sì visto che abbiamo modificato un nodo fortemente caratteristico, che è lo stato pozzo.

**Confronto 1**

Confronto il Tomita15 con un suo duplicato in cui dallo Stato 3, con la lettera a, andiamo allo stato 0.

Il punteggio di similarità va diventando significativamente sempre più piccolo al diminuire del fattore di terminazione epsilon:

**Epsilon ||| Similarità**

0.1 0.617105

0.01 ﷒﷒ 0.0679213

0.001 0.0067411

0.0001 0.000669046

… …

0.000000001 ﷒ 7.15622e-09

… …

Ne deduciamo che la similarità tende a zero.

Cambiare una sola transizione da un automa può farlo diventare completamente diverso dalla sua versione originale?

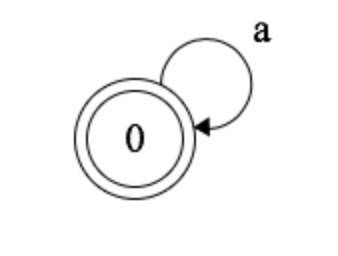
Innanzitutto questo è un automa con pochi stati e poche transizioni.

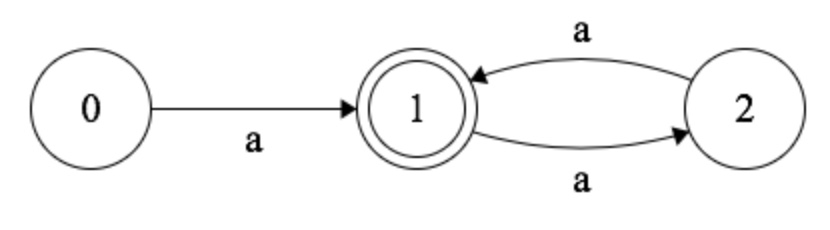
Facendo arrivare una transizione all’interno dello stato 0, cambiamo l’insieme dei suoi vicini entranti, che nel Tomita15 era nullo, e adesso ha lo stato 3 come elemento. Lo stato 0 ha tre collegamenti con tre su quattro degli altri stati dell’automa, per cui quando si confronterà la similarità entrante dei suddetti stati, occorrerà cercare il matching ottimo per la similarità del loro vicinato entrante, e cioè proprio dello stato zero. E lo stato zero dei due automi confrontati tra loro avrà sempre un punteggio di similarità entrante pari a zero, perché nel tomita originale non ci sono transizioni entranti nello stato 0, nella versione modificata ce n’è una.

Poi l’unico stato non raggiunto dallo stato 0, è lo stato 3, al quale abbiamo variato l’insieme del vicinato uscente.

Mi pare che sia per queste ragioni che il punteggio di similarità vada convergendo a zero.

**Confronto 2**

****

****

Confrontando questi due automi, il punteggio di similarità va diventando significativamente sempre più piccolo al diminuire del fattore di terminazione epsilon:

**Epsilon ||| Similarità**

0.1 0.316406

0.01 0.0316764

0.001 0.00317121

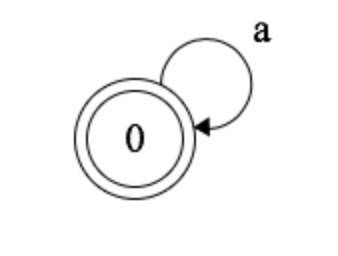
0.0001 0.000317479

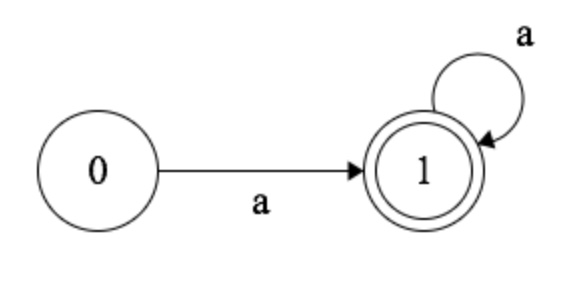
… …

Ne deduciamo che la similarità tende a zero.

Sembra essere un risultato ragionevole dato che nessuno stato del secondo automa ha una transizione che torna in sé stesso. Eppure mi aspettavo un punteggio di similarità che tendesse più velocemente a zero al variare di epsilon.

**Confronto 3**

****

****

**Epsilon ||| Similarità**

0.1 0.237305

0.01 0.0237573

… …

0.000000001 2.39457e-09

Consideriamo epsilon = 0.1 può starci che la similarità sia vicina al 25%, dato che il matching è tra lo stato 0 e lo stato 1 dei due automi, dove però lo stato 1 ha un vicino entrante che non sia sé stesso, dimezzando per questa ragione la similarità entrante. Quello che mi ha sorpreso inizialmente è che a parità di epsilon, il confronto 2 scritto più sopra ha un punteggio di similarità più alto. Ma mi sono fatto ingannare dalla vista di una transizione che fa andare nello stesso stato, presente nel caso di questo ultimo confronto numero tre.

Guardiamo le matrici di similarità per i due confronti, con epsilon = 0.1

Matrice similarità confronto 2

Matrice similarità confronto 3

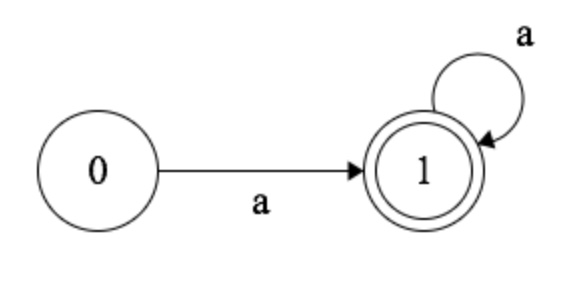
Lo stato 1 nei due automi subject dei confronti, è strutturalmente identico a quello dell’altro: hanno due transizioni entranti e una uscente (nel primo la transizione va in un altro stato, nel secondo la transizione torna nello stesso stato). Infatti il punteggio di similarità inter-nodo è identico.

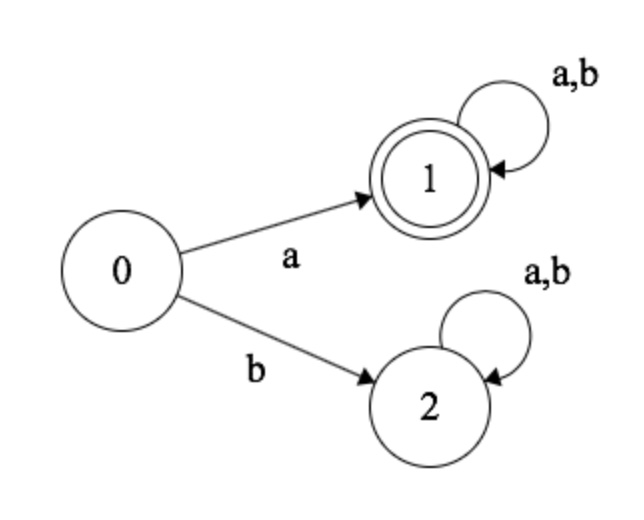
Lo stato 0 dell’automa reference è però più simile allo stato 2 del confronto 2, il quale ha una sola transizione entrante anziché due.

La differenza nel numero di stati non è penalizzante in questa implementazione perché ricordiamo che il valore di similarità globale è così ricavato:

dove

**Confronto 4**

****



**Epsilon ||| Similarità**

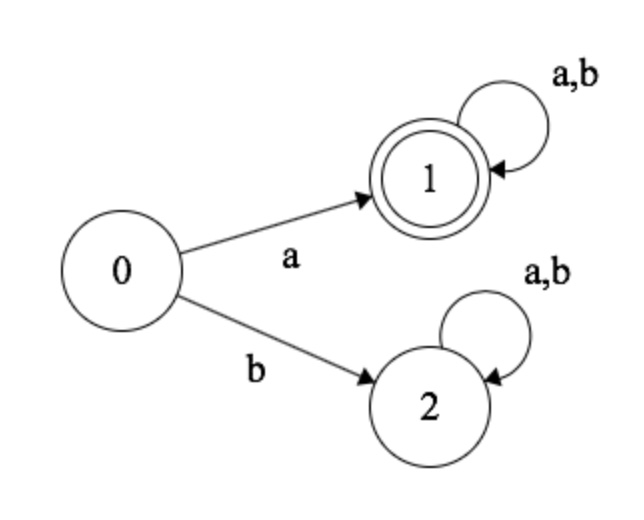
0.1 0.84375

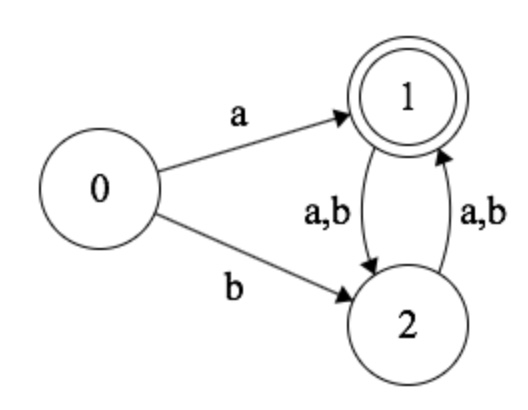
0.01 0.692688

0.001 0.66928

0.0001 0.666929

**Confronto 5**





**Epsilon ||| Similarità**

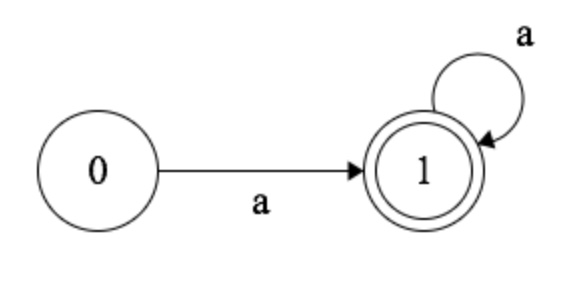
0.0001 1

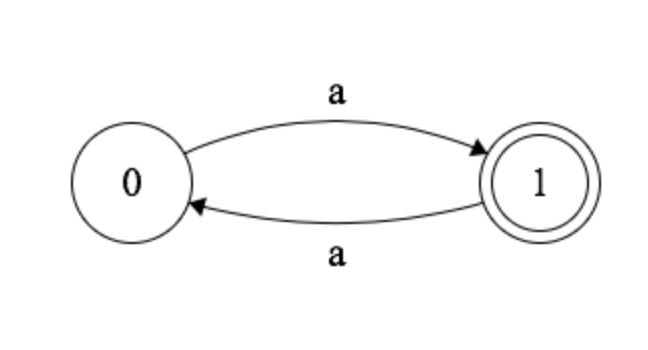
Esempio che mi sembra significativo di linguaggio diverso ma struttura identica.

Anche se mi chiedo: una transizione che fa tornare lo stato in sé stesso, è una transizione entrante ordinaria? Non dobbiamo considerarla in maniera diversa in questi casi? Forse no.

Sembra però mancare il concetto speculare al “nodo sorgente” ossia nodo senza alcuna transizione in ingresso ma solo in uscita. Un nodo pozzo non deve essere considerato come nodo terminazione? Senza quindi transizioni in uscita (anche se tecnicamente ha delle transizioni che portano su sé stesso)?

**Confronto 6**

****

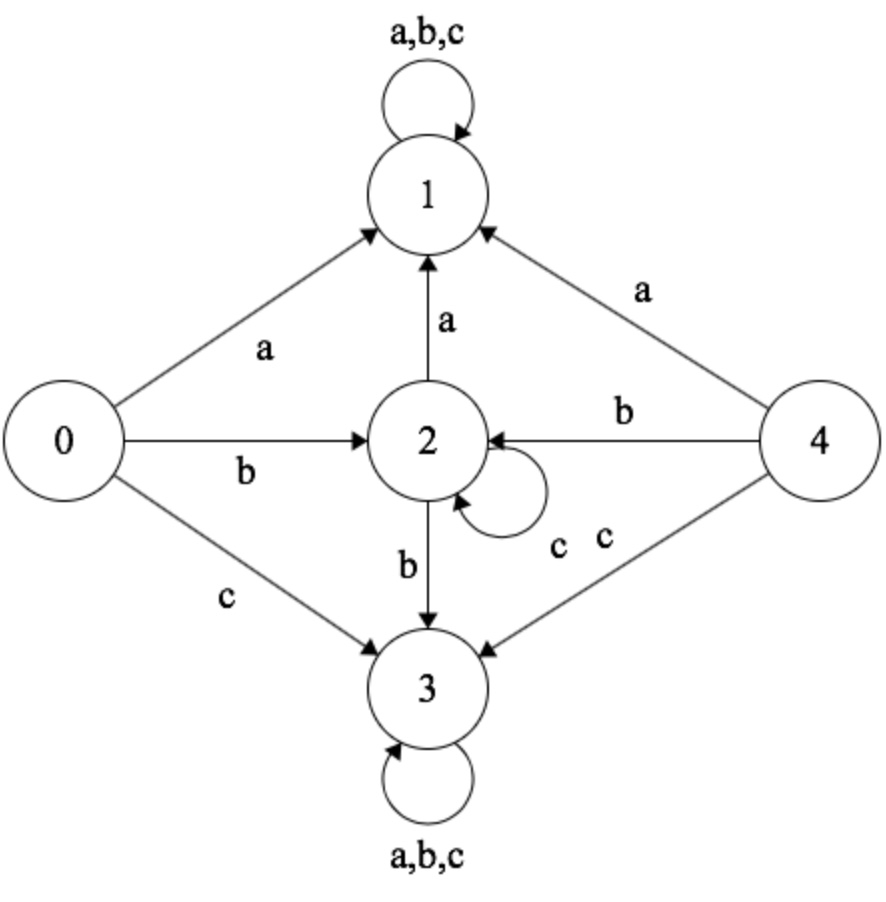


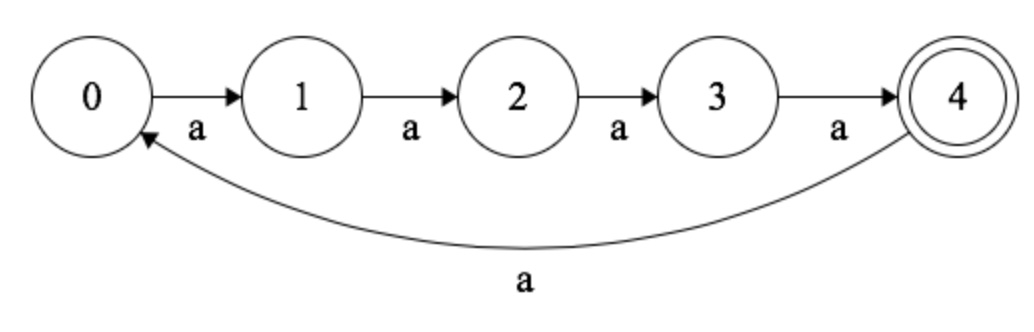
**Epsilon ||| Similarità**

0.0001 1

Come confronto 5.

**Confronto 7**

****

****

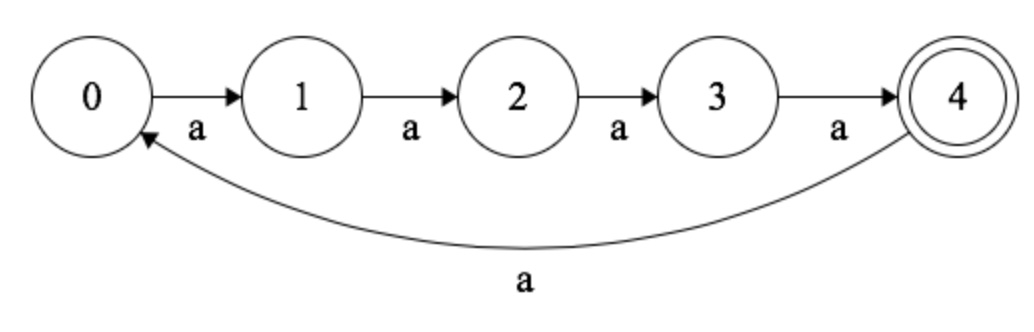
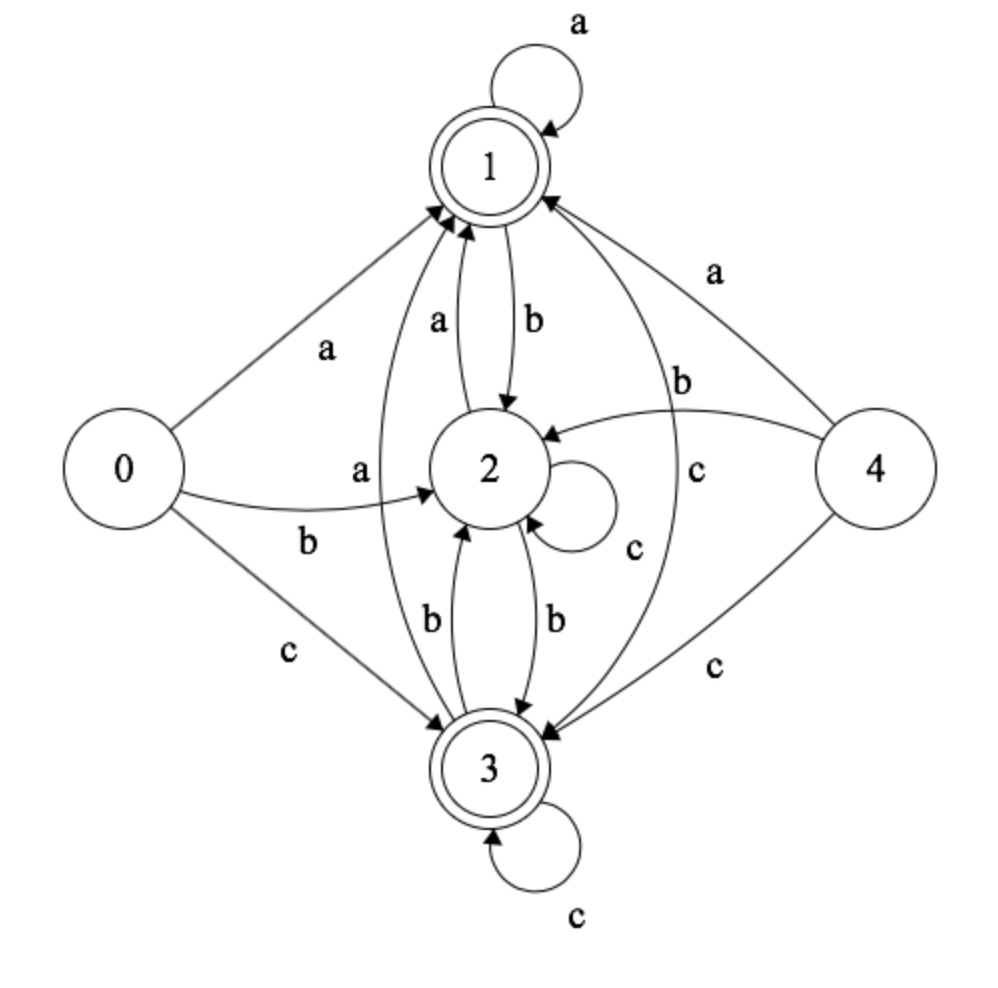
**Epsilon ||| Similarità**

0.1 0.0876194

0.01 0.00835603

Già per valori alti di epsilon il punteggio di similarità è molto basso: questo accade perché nel reference dfa abbiamo due stati “sorgente” senza alcuna transizione entrante, mentre il subject dfa ha per tutti gli stati almeno una transizione entrante.

**Confronto 8**

****

**Epsilon ||| Similarità**

0.1 0.0145185

0.01 0.000813899

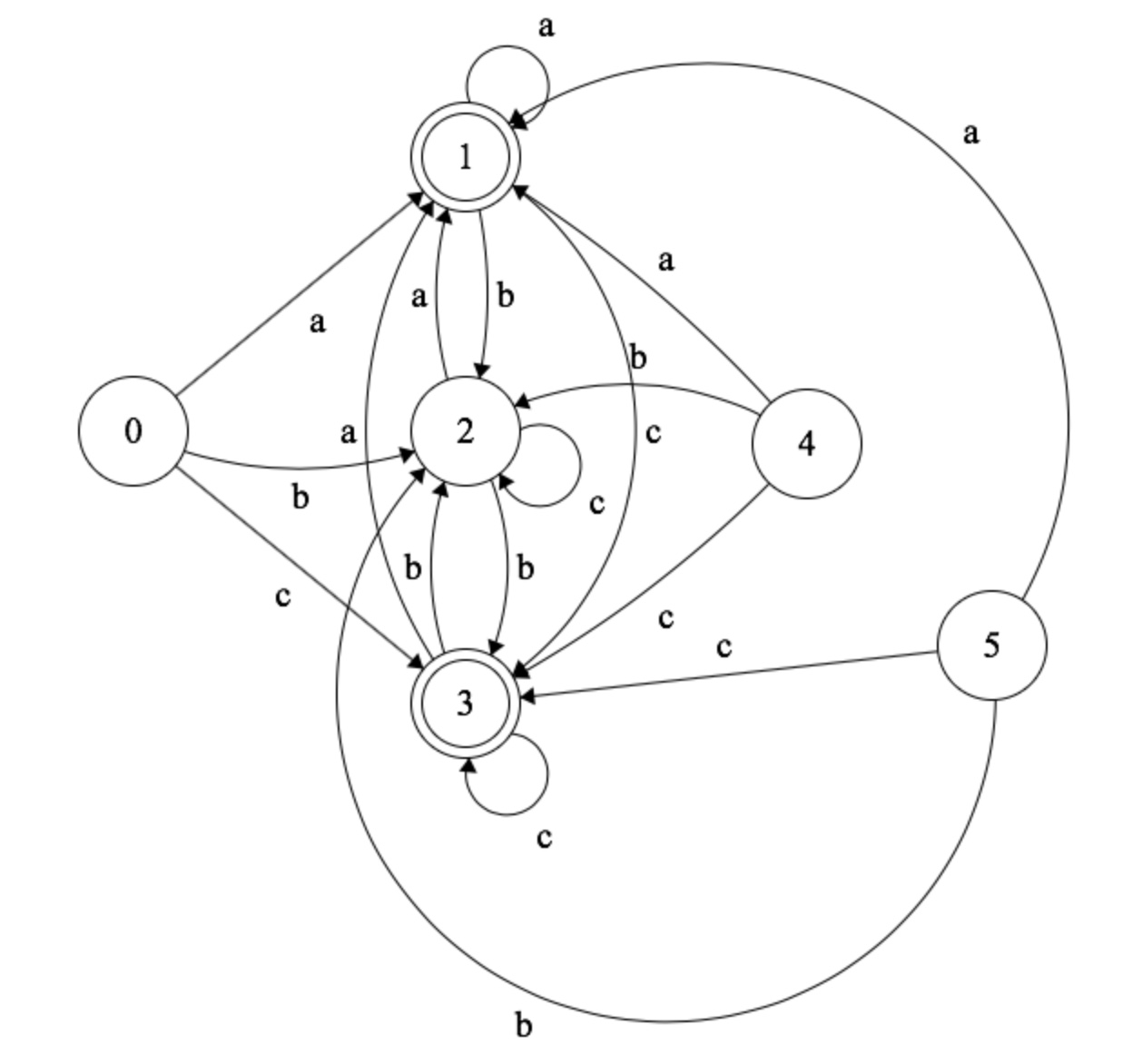
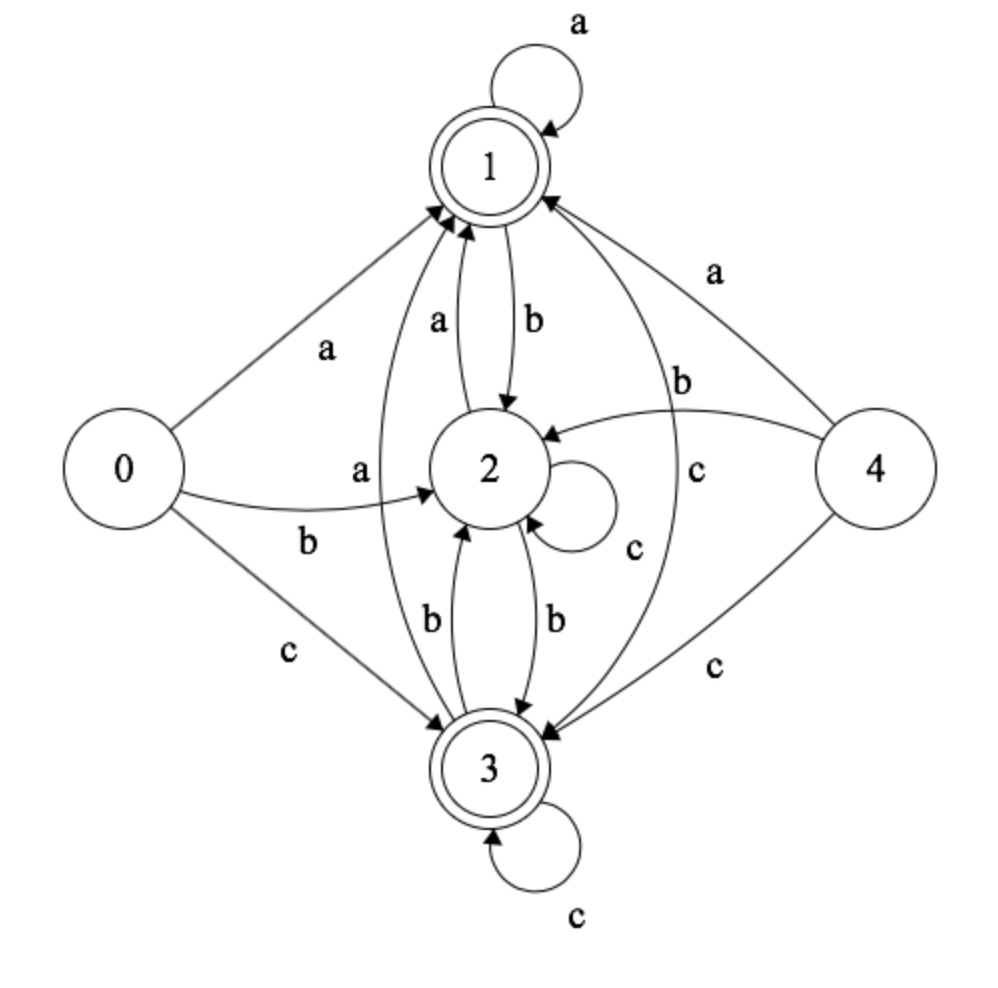
Caso più estremo del confronto 7: per epsilon pari a 0.1 gli automi si somigliano strutturalmente dell’ 1%.

Non solo come prima non c’è nessuno stato nel secondo automa che non abbia alcuna transizione in ingresso. In più ogni stato del primo automa ha 3 transizioni distinte in uscita, nel secondo automa ciascuno stato ha una sola transizione.

Inoltre qui il minimo tra il numero degli stati dei due automi, il coefficiente n che smorza il punteggio di similarità globale, è pari al numero degli stati di entrambi gli automi.

Quindi secondo questa misura di similarità, due automi sono totalmente diversi quando uno possiede un nodo sorgente e l’altro no, e il numero di transizioni entranti e uscenti è uniformemente distante confrontando tutti gli stati di un automa con tutti gli stati dell’altro.

**Confronto 9**

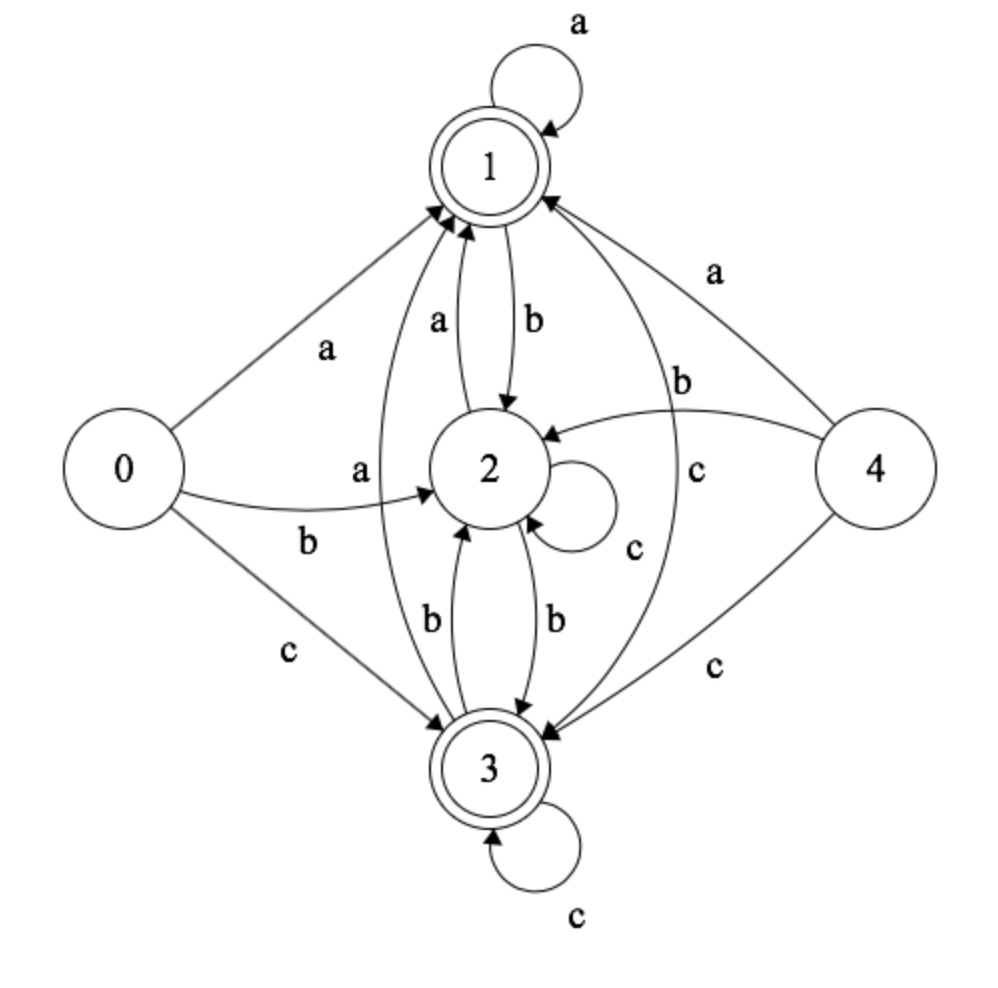
****

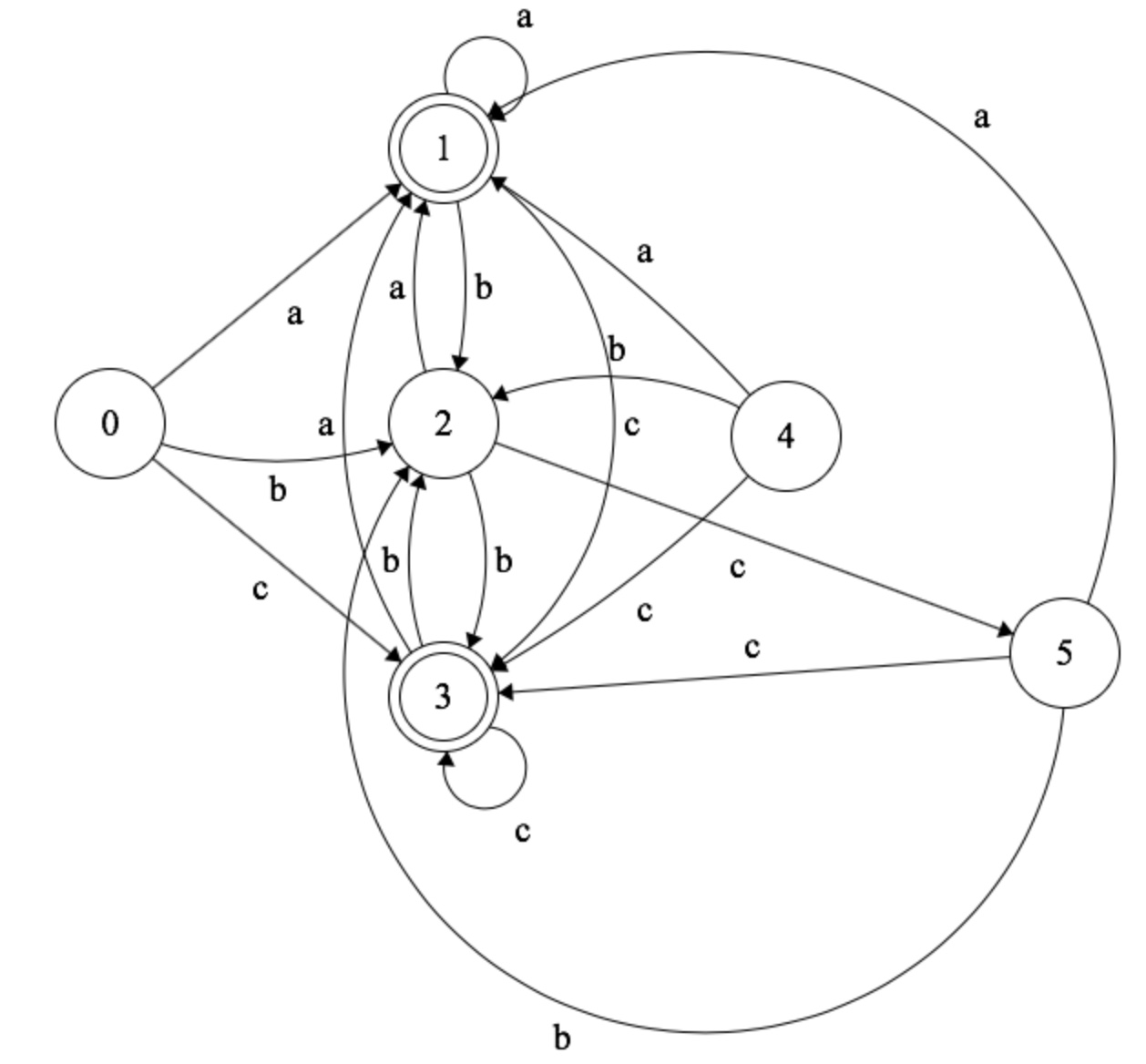
**Epsilon ||| Similarità**

0.0001 0.600409

0.00001 0.600041

**Confronto 10**

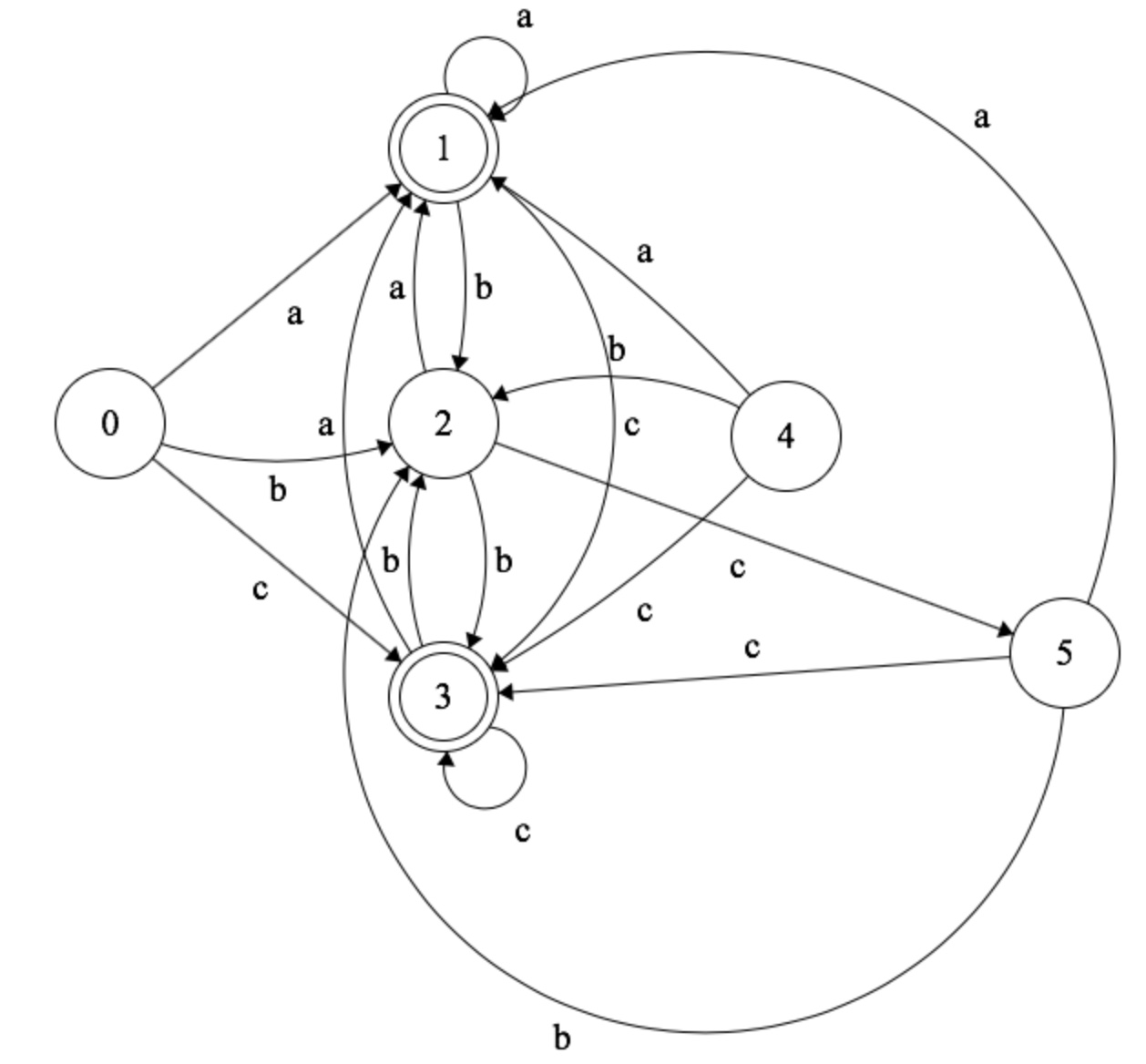
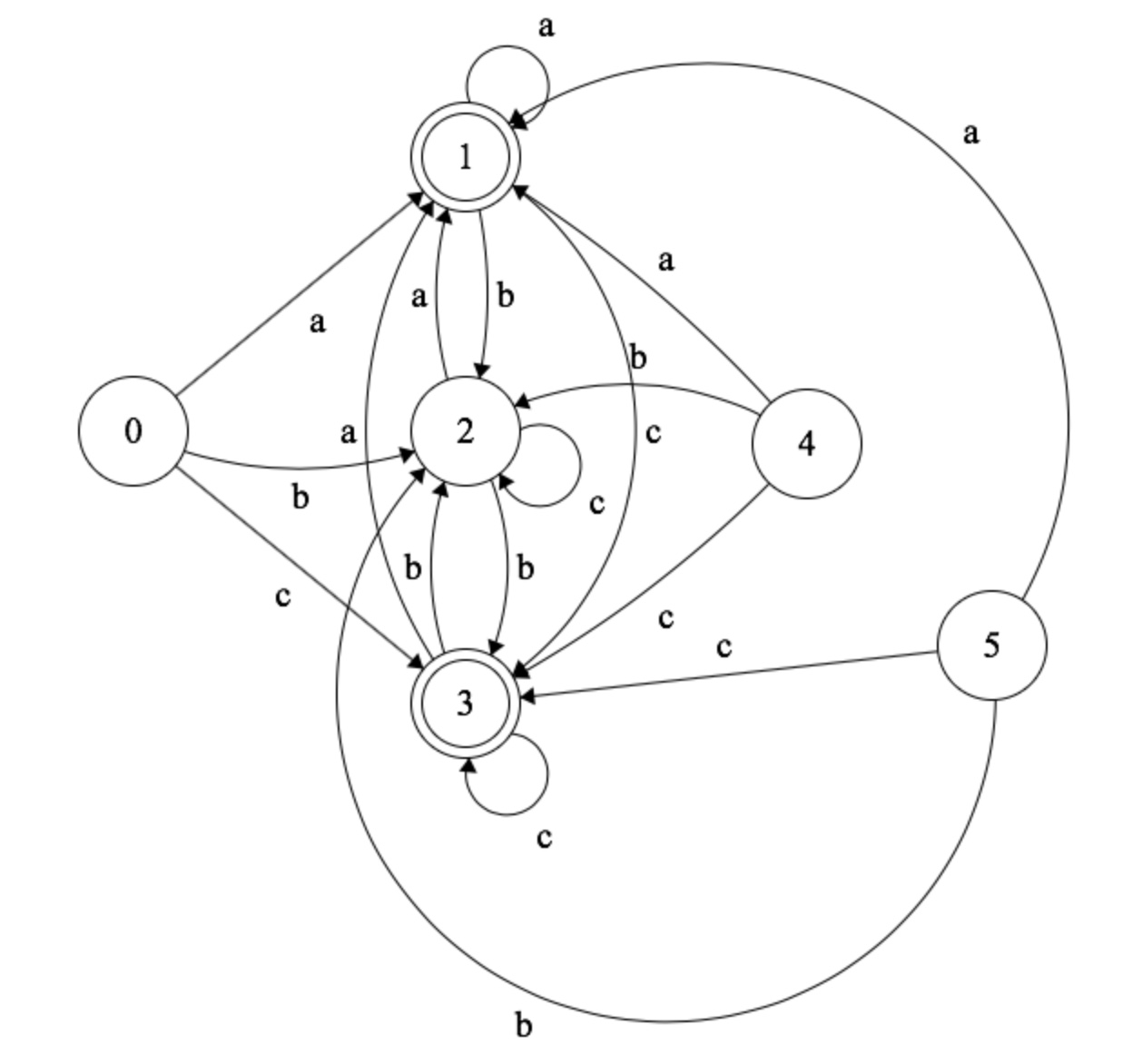
****

****

**Epsilon ||| Similarità**

0.0000000001 0.595699

**Confronto 11**

****

**Epsilon ||| Similarità**

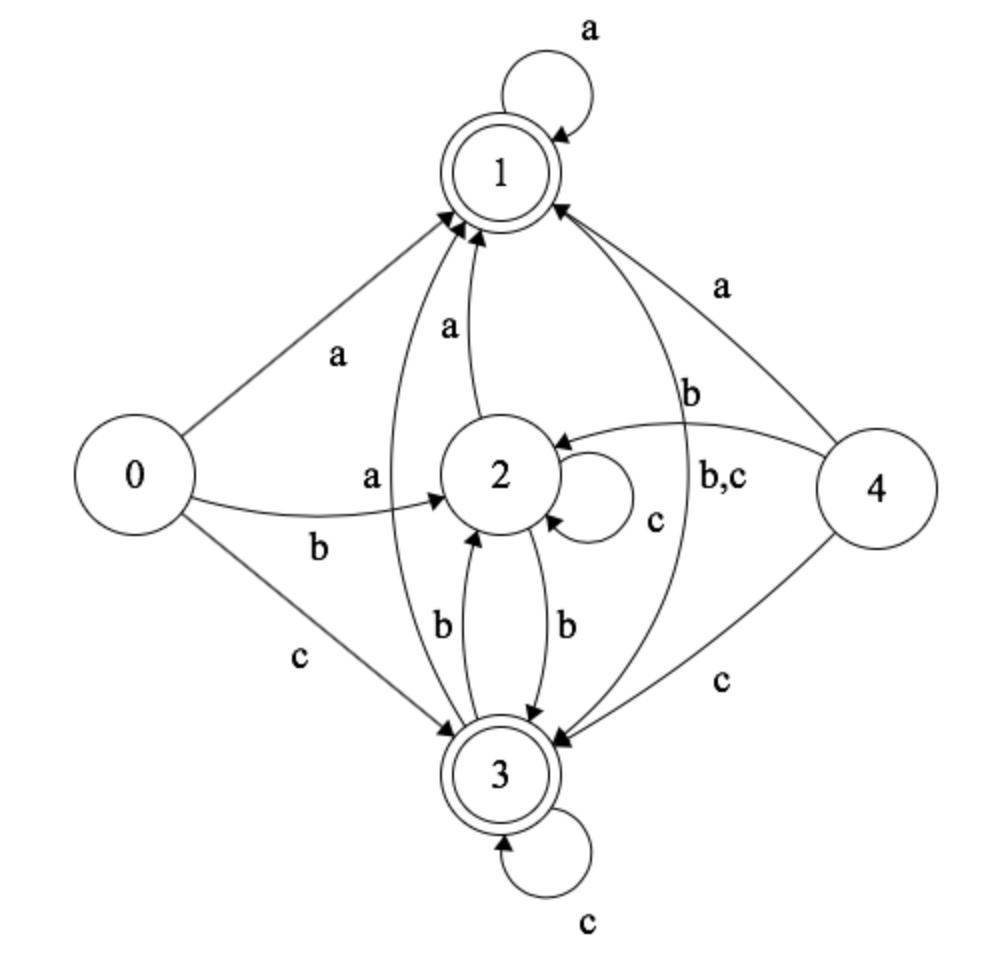
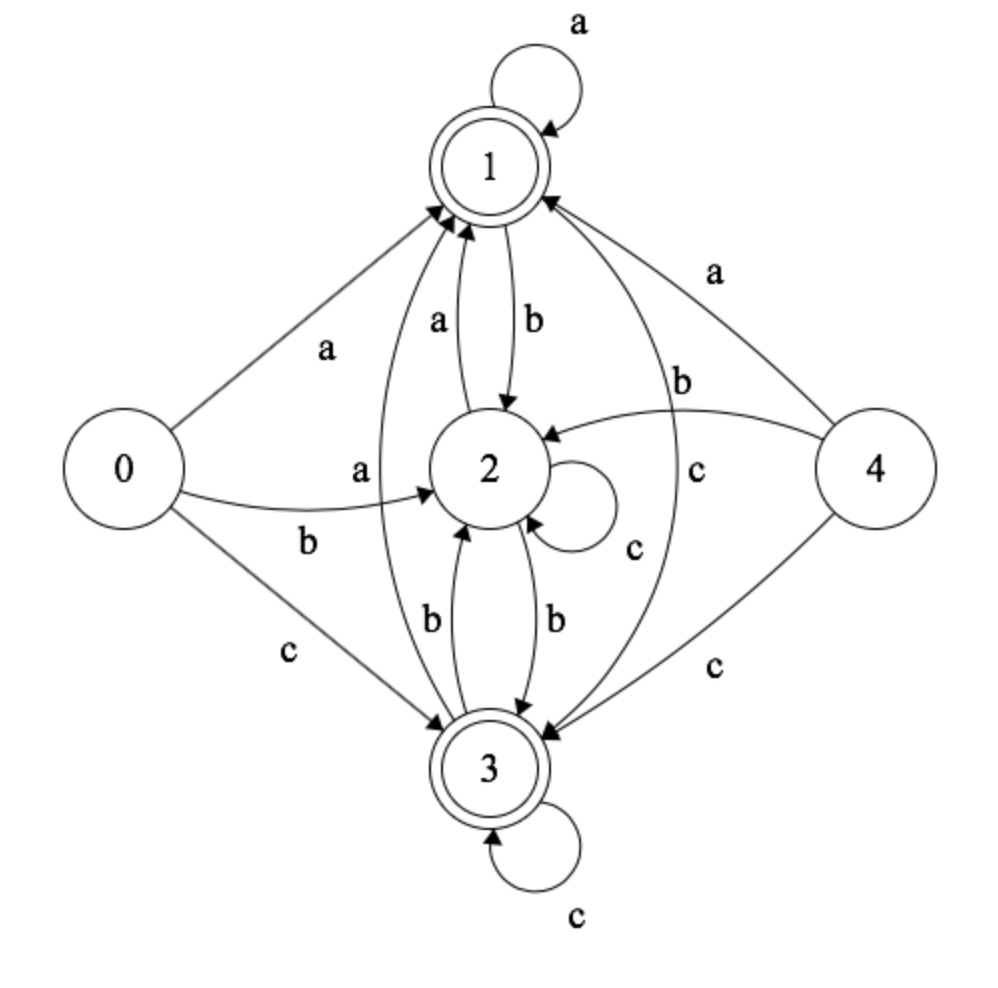
0.1 0.811825

0.01 0.584368

0.001 0.545025

0.0001 0.540999

**Confronto 12**

****

Lo stato 1 non è più connesso allo stato 2 tramite b, nella versione modificata di destra la transizione da 1 con b va in 3.

**Epsilon ||| Similarità**

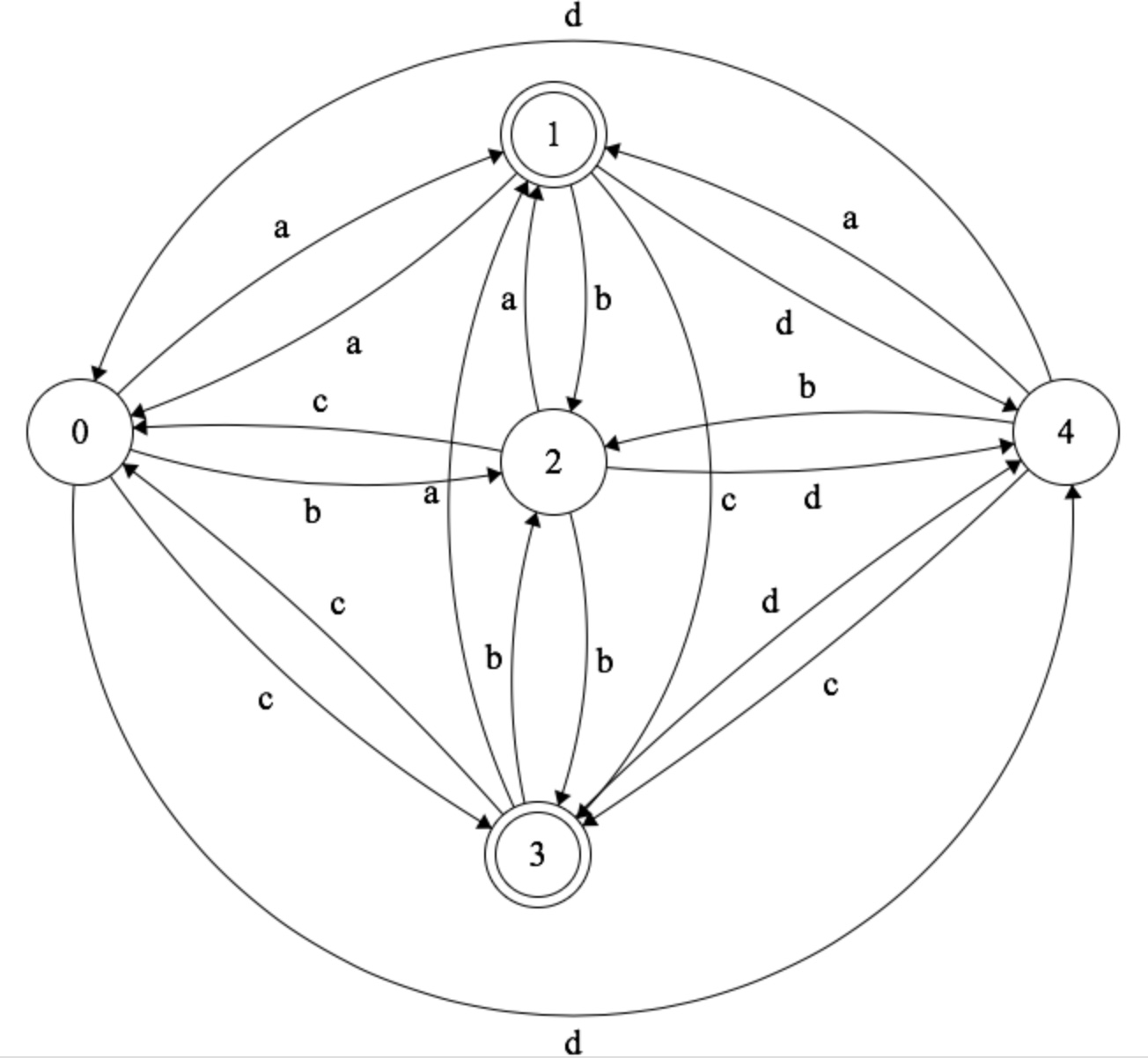
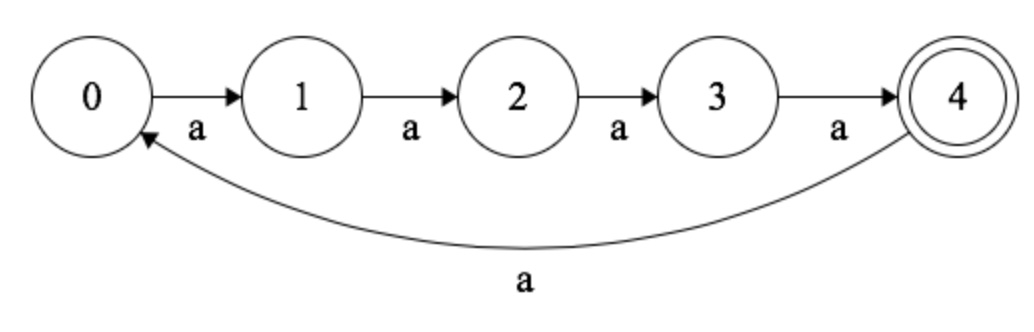
0.1 0.892889

0.01 0.66627

0.001 0.63378

0.0001 0.630688

**Confronto 13**

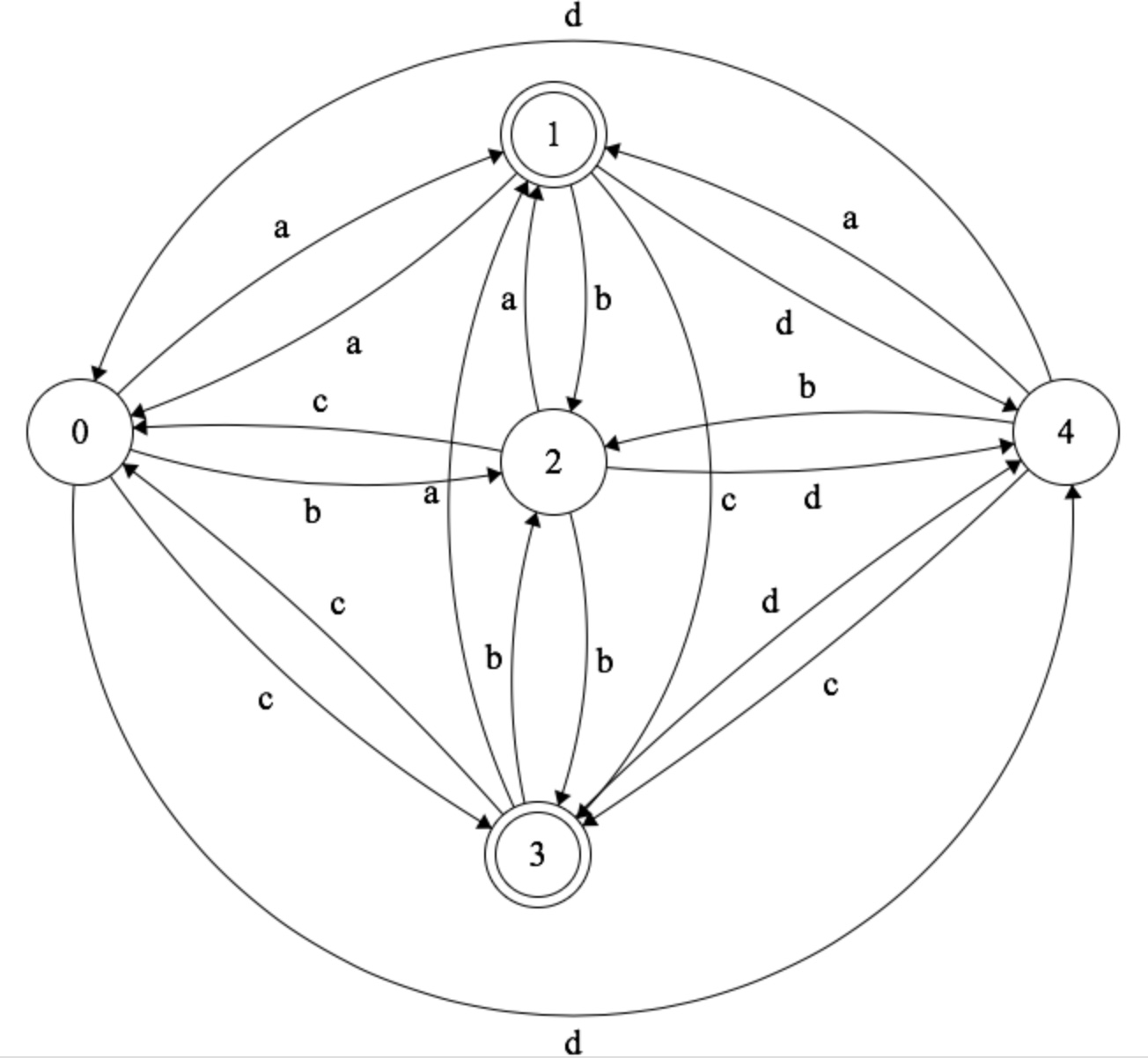
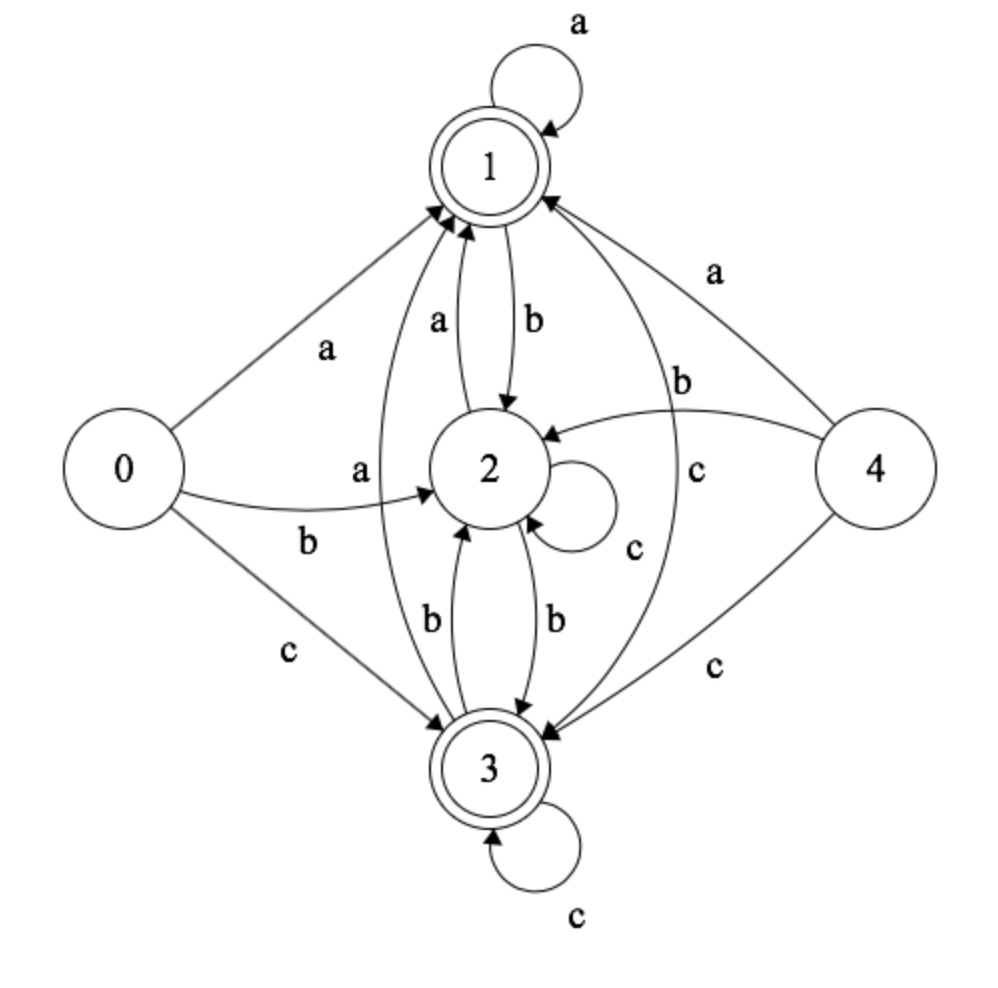
****

**Epsilon ||| Similarità**

0.1 0.015625

0.01 0.000976562

**Confronto 14**

****

**Epsilon ||| Similarità**

0.1 0.17349

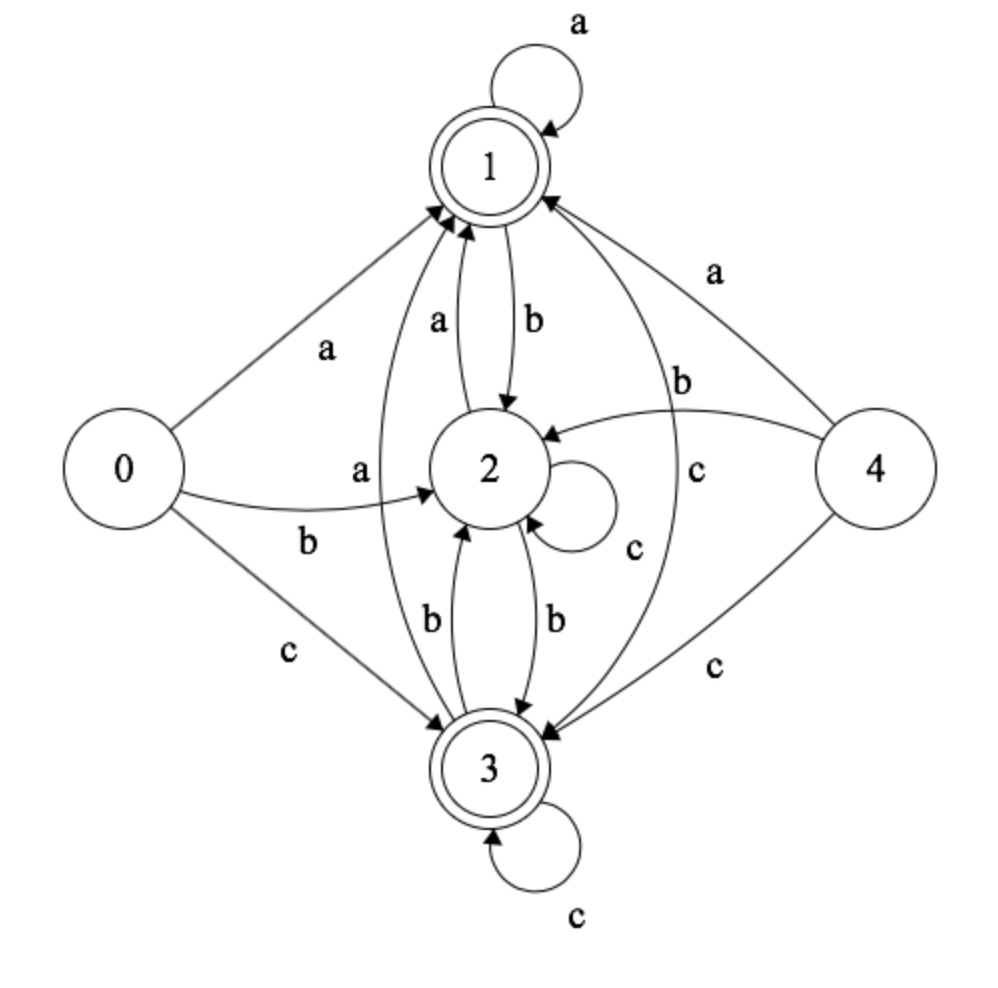
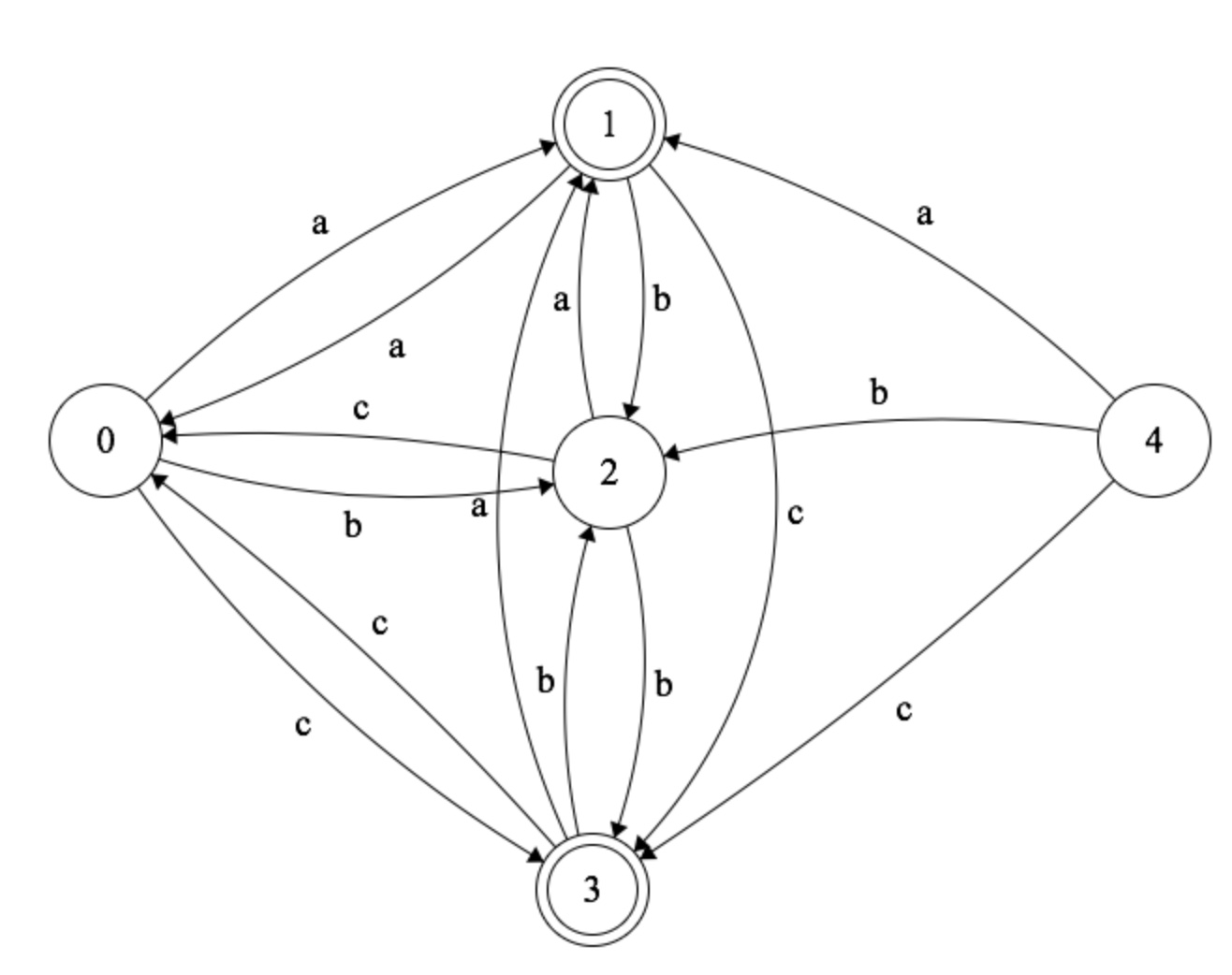
0.01 0.0185517

0.001 0.00198377

0.0001 0.000212128

0.00001 1.64819e-05

**Confronto 15**

****

**Epsilon ||| Similarità**

0.1 0.684733

0.01 0.340158

0.001 0.30083

0.0001 0.29711

0.00001 0.296765