INDICE

1. Automi a Stati Finiti

Un automa è il modello matematico di un sistema con input ed output discreti. Il sistema può trovarsi in una fra n configurazioni diverse, dette **STATI**.
Lo stato rappresenta una "condizione" in cui il sistema si trova, e sommarizza la storia precedente del sistema in termini di input ricevuti.
In un automa a stati finiti, la conoscenza dello stato in cui si trova il sistema è necessaria per determinare il comportamento del sistema a fronte di input successivi.
A fronte di un nuovo input, il sistema transita in un nuovo stato, e questo è rappresentato graficamente nel modello con un arco orientato dallo stato di partenza a quello di arrivo, etichettato con il valore dell'input.

La definizione formale di automa a stati finiti è la seguente:

DEF. Un *automa a stati finiti* è una quintupla $(Q, \Sigma, \Delta, \delta, q_0)$ dove Q è un insieme finito di stati Σ è un alfabeto finito di simboli, q_0 è lo stato iniziale, $\Delta \subseteq Q$ è il set di stati finali, e δ è la funzione di transizione $Q \times \Sigma \rightarrow Q$ ($Q \times \Sigma$ è il prodotto cartesiano, ovvero l'insieme delle coppie (q,a)); $\delta(q,a)$ rappresenta uno **stato** raggiunto dall'automa, per ogni stato di partenza q e simbolo di ingresso a.

La definizione appena fornita in realtà si riferisce ad un sottoinsieme di automi a stati finiti, gli automi *deterministici*. Per questa classe di automi, la funzione $\delta(q,a)$ ha un unico valore, cioè, non è ambigua.

Simbologia adottata:

1. se Q è un insieme di **stati**, q_i indica uno stato, q_0 indica uno stato iniziale
2. Σ è l'**alfabeto di ingresso** (nel caso di sistemi binari, $\Sigma = \{0, 1\}$)
3. δ indica una **funzione di transizione** $\delta : Q \times \Sigma \rightarrow Q$
4. Δ è il set degli stati **finali**, o di **accettazione** $\Delta \subset Q$

Il comportamento di un automa può essere rappresentato mediante un **grafo** in cui gli archi rappresentano le transizioni fra stati, i nodi, denotati da cerchi, rappresentano gli stati. Gli stati finali sono evidenziati da cerchi doppi. I simboli sugli archi sono gli input del sistema, ed appartengono all'alfabeto Σ. Alternativamente, un automa si può rappresentare mediante una **tabella delle transizioni, o stati futuri**, definiti mediante la funzione δ.

2. Automi a stati finiti con output

Il comportamento di un circuito sequenziale può essere modellato mediante un particolare tipo di automi a stati finiti : gli automi deterministici con output. Questi automi vengono chiamati macchine di *Moore* o di *Mealy.*, a seconda che l'output sia associato agli stati, o alle transizioni fra stati.

3. Automa di Moore

Def. Un automa (o macchina) di Moore è una sestupla $<Q, \Sigma, \Delta, \delta, \lambda, q0>$, dove:

1. Q = insieme finito degli stati: "insieme degli stati";
2. Σ = insieme finito dei simboli di ingresso: "alfabeto di input";
3. Δ = insieme finito dei simboli di uscita: "alfabeto di output";
4. $\delta: Q \times \Sigma \rightarrow Q$ funzione di transizione;
5. $\lambda: Q \rightarrow \Delta$ funzione di uscita;
6. $q0$ stato iniziale.

Nel nostro caso si ha un automa che riconosce i numeri congrui a 0 modulo 7.
Per costruirlo si è pensato di ragionare come segue:

- dato un numero $m \in N$ in base 10, lo si converte in base 2;
- si percorre il grafo generato da δ partendo dallo stato $q0 \in Q$;
- studiamo la sequenza finita, denotata con num[i], di bit di n, da sinistra verso destra;
- ad ogni passo percorriamo l'arco denotato con il simbolo letto num[i].

Quindi, sia n la sequenza di bit finora studiata, si avrà la seguente evoluzione (dove $x \in N$):

se stiamo nello stato q0 (cioè n è congruo a 0 modulo 7, $n \equiv 0\ (7)$) allora
se si legge 0 si resta in q0 (poiché "n0" equivale a $2n \equiv 0\ (7)$)
se si legge 1 si passa in q1 (poiché "n1" equivale a $(2n+1) \equiv 1\ (7)$)

se stiamo nello stato q1 (cioè n è congruo a 1 modulo 7, $n \equiv 1\ (7)$) allora
se si legge 0 si passa in q2 (poiché "n0" equivale a $2n = 2(7x+1) \equiv 2\ (7)$)
se si legge 1 si passa in q3 (poiché "n1" equivale a $(2n+1) = 2\ (7x+1) + 1 \equiv 3\ (7)$)

se stiamo nello stato q2 (cioè n è congruo a 2 modulo 7, $n \equiv 2\ (7)$) allora
se si legge 0 si passa in q4 (poiché "n0" equivale a $2n = 2(7x+2) \equiv 4\ (7)$)
se si legge 1 si passa in q5 (poiché "n1" equivale a $(2n+1) = 2(7x+2) +1 \equiv 5\ (7)$)

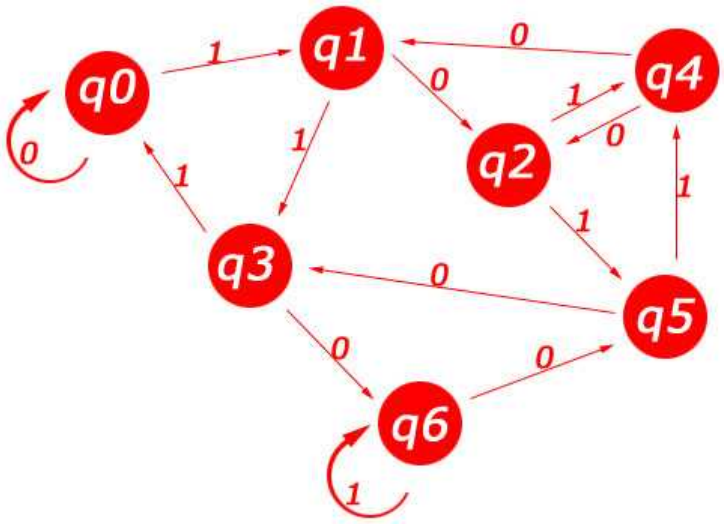
se stiamo nello stato q3 (cioè n è congruo a 3 modulo 7, $n \equiv 3\ (7)$) allora
se si legge 0 si passa in q6 (poiché "n0" equivale a $2n = 2(7x+3) \equiv 6\ (7)$)
se si legge 1 si passa in q0 (poiché "n1" equivale a $(2n+1) = 2(7x+3) +1 \equiv 0\ (7)$)

se stiamo nello stato q4 (cioè n è congruo a 4 modulo 7, $n \equiv 4\ (7)$) allora
se si legge 0 si passa in q1 (poiché "n0" equivale a $2n = 2(7x+4) \equiv 1\ (7)$)
se si legge 1 si passa in q2 (poiché "n1" equivale a $(2n+1) = 2(7x+4) + 1 \equiv 2\ (7)$)

se stiamo nello stato q5 (cioè n è congruo a 5 modulo 7, $n \equiv 5\ (7)$) allora
se si legge 0 si passa in q3 (poiché "n0" equivale a $2n = 2(7x+5) \equiv 3\ (7)$)
se si legge 1 si passa in q4 (poiché "n1" equivale a $(2n+1) = 2(7x+5) +1 \equiv 4\ (7)$)

se stiamo nello stato q6 (cioè n è congruo a 6 modulo 7, $n \equiv 6\ (7)$) allora
se si legge 0 si passa in q5 (poiché "n0" equivale a $2n = 2(7x+6) \equiv 5\ (7)$)
se si legge 1 si resta in q6 (poiché "n1" equivale a $(2n+1) = 2(7x+6) +1 \equiv 6\ (7)$)

Il corrispondente automa sarà quindi:



Dove q0 è lo stato iniziale e ha come output associato 0
(cioè termineranno nello stato q0 i numeri congrui a 0 modulo 7)
Gli output associati agli altri stati seguono dalla definizione di λ :

stati	**output**
q0	0
q1	1
q2	2
q3	3
q4	4
q5	5
q6	6

Analogamente possiamo ricavare δ come segue:

	0	**1**
q0	q0	q1
q1	q2	q3
q2	q4	q5
q3	q6	q0
q4	q1	q2
q5	q3	q4
q6	q5	q6

4. Algoritmo in C che descrive il comportamento di un automa di Moore

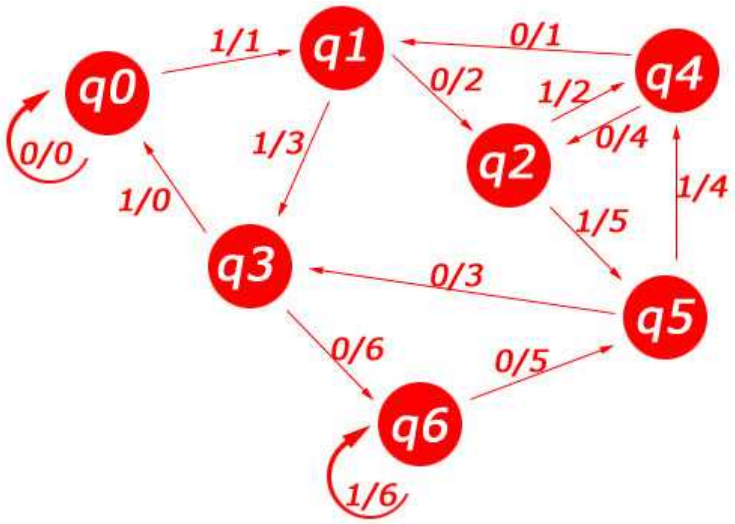
```
#include <stdio.h>
#define dim 20

int main (void)
{ int a,j,m,n,i=0,c=0;
  int num[dim];
  printf("\nSIMULAZIONE DI UN AUTOMA A STATI FINITI DI MOORE\n");
  printf("\nViene calcolato il resto modulo 7 di un numero intero\n");
  printf("\nInserire un numero intero (base dieci):  ");
  scanf("%d",&m);
  n=m;
  printf("\n\n");
  while (n>0)
  { if (i>=dim)  {printf("numero fuori range\n\n"); return(0);  };
    num[i]=n%2;
    i++;
    n/=2; }
  printf("- trasformato in base due diventa:  ");
  for (j=i-1;j>=0;j--) printf("%d ",num[j]);
  printf("\n\nL'evoluzione dell'automa con stato iniziale q0 e' la seguente :
\n\n");
  for (j=i-1;j>=0;j--)
    { a=num[j];
      if (c==0)   if (a==0) {printf (" ( q0,0 ) -> q0\n");  c=0; }
                  else      {printf (" ( q0,1 ) -> q1\n");  c=1; }
      else
      if (c==1)   if (a==0) {printf (" ( q1,0 ) -> q2\n");  c=2; }
                  else      {printf (" ( q1,1 ) -> q3\n");  c=3; }
      else
      if (c==2)   if (a==0) {printf (" ( q2,0 ) -> q4\n");  c=4; }
                  else      {printf (" ( q2,1 ) -> q5\n");  c=5; }
      else
      if (c==3)   if (a==0) {printf (" ( q3,0 ) -> q6\n");  c=6; }
                  else      {printf (" ( q3,1 ) -> q0\n");  c=0; }
      else
      if (c==4)   if (a==0) {printf (" ( q4,0 ) -> q1\n");  c=1; }
                  else      {printf (" ( q4,1 ) -> q2\n");  c=2; }
      else
      if (c==5)   if (a==0) {printf (" ( q5,0 ) -> q3\n");  c=3; }
                  else      {printf (" ( q5,1 ) -> q4\n");  c=4; }
      else
      if (c==6)   if (a==0) {printf (" ( q6,0 ) -> q5\n");  c=5; }
                  else      {printf (" ( q6,1 ) -> q6\n");  c=6; }
    }
  printf ("\nIl resto modulo 7 di %d  e'  %d\n\n",m,c);
  printf ("Programma terminato.\n\n");
  return(0);
}
```

5. Automa di Mealy

Def. Un automa (o macchina) di Mealy è una sestupla <Q, Σ, Δ, δ, λ, q0>, dove:

1. Q = insieme finito degli stati: "insieme degli stati";
2. Σ = insieme finito dei simboli di ingresso: "alfabeto di input";
3. Δ = insieme finito dei simboli di uscita: "alfabeto di output";
4. δ: Q x Σ → Q funzione di transizione;
5. λ: Q x Σ → Δ funzione di output;
6. q0 stato iniziale.

La costruzione dell'automa di Mealy, per quanto detto precedentemente, è molto semplice, in quanto basterà associare gli output non allo stato ma all'arco di transizione.
Nello specifico si avrà:



Le funzioni di transizione δ sarà la stessa di quella di Moore (come vista dettagliatamente) mentre la funzione di transizione λ sarà quindi:

	0	**1**
q0	0	1
q1	2	3
q2	4	5
q3	6	0
q4	1	2
q5	3	4
q6	5	6

6. Algoritmo in C che descrive il comportamento di un automa di Mealy

```
#include <stdio.h>
#define dim 20

int main (void)
{ int a,j,m,n,i=0,c=0;
  int num[dim];
  printf("\n  SIMULAZIONE DI UN AUTOMA A STATI FINITI DI MEALY\n\n");
  printf("\n  Viene calcolato il resto modulo 7 di un numero intero\n");
  printf("\n  inserire un numero intero (base dieci):  ");
  scanf("%d",&m);
  n=m;
  printf("\n\n");
  while (n>0)
  { if (i>=dim)  {printf("numero fuori range\n\n"); return(0);  };
    num[i]=n%2;
    i++; n/=2; }
  printf("- trasformato in base due diventa:  ");
  for (j=i-1;j>=0;j--) printf("%d ",num[j]);
  printf("\n\nL'evoluzione dell'automa con stato iniziale q0 e' la seguente :");
  printf("\n\n  Legenda:\n  (stato vecchio)  --[ simbolo letto, output
ottenuto ]-->  (stato nuovo) )\n\n");
  for (j=i-1;j>=0;j--)
    { a=num[j];
      if (c==0)   if (a==0) {printf ("  (q0)  --[0, 0]->  (q0)\n");  c=0; }
                  else      {printf ("  (q0)  --[1, 1]->  (q1)\n");  c=1; }
      else
      if (c==1)   if (a==0) {printf ("  (q1)  --[0, 2]->  (q2)\n");  c=2; }
                  else      {printf ("  (q1)  --[1, 3]->  (q3)\n");  c=3; }
      else
      if (c==2)   if (a==0) {printf ("  (q2)  --[0, 4]->  (q4)\n");  c=4; }
                  else      {printf ("  (q2)  --[1, 5]->  (q5)\n");  c=5; }
      else
      if (c==3)   if (a==0) {printf ("  (q3)  --[0, 6]->  (q6)\n");  c=6; }
                  else      {printf ("  (q3)  --[1, 0]->  (q0)\n");  c=0; }
      else
      if (c==4)   if (a==0) {printf ("  (q4)  --[0, 1]->  (q1)\n");  c=1; }
                  else      {printf ("  (q4)  --[1, 2]->  (q2)\n");  c=2; }
      else
      if (c==5)   if (a==0) {printf ("  (q5)  --[0, 3]->  (q3)\n");  c=3; }
                  else      {printf ("  (q5)  --[1, 4]->  (q4)\n");  c=4; }
      else
      if (c==6)   if (a==0) {printf ("  (q6)  --[0, 5]->  (q5)\n");  c=5; }
                  else      {printf ("  (q6)  --[1, 6]->  (q6)\n");  c=6; }
    }
  printf ("\nIl resto modulo 7 di %d  e'  %d\n",m,c);
  printf ("\nProgramma terminato.\n\n");
  return(0);
}
```

7. Automa di Mealy e di Moore a confronto

- Le macchine di Mealy hanno, per ogni arco, un simbolo di entrata e uno di uscita

  Nelle di macchine di Moore l'uscita è invece già codificata nel valore dello stato in cui si trova la macchina, ovvero la funzione di uscita dipende solo da Q, anziché da Q x Σ

- In una macchina di Moore si ha una **uscita dopo che è avvenuta la transizione** e questa uscita dipende solo dal nuovo stato della macchina

  In una macchina di Mealy si ha una uscita durante ciascuna transizione

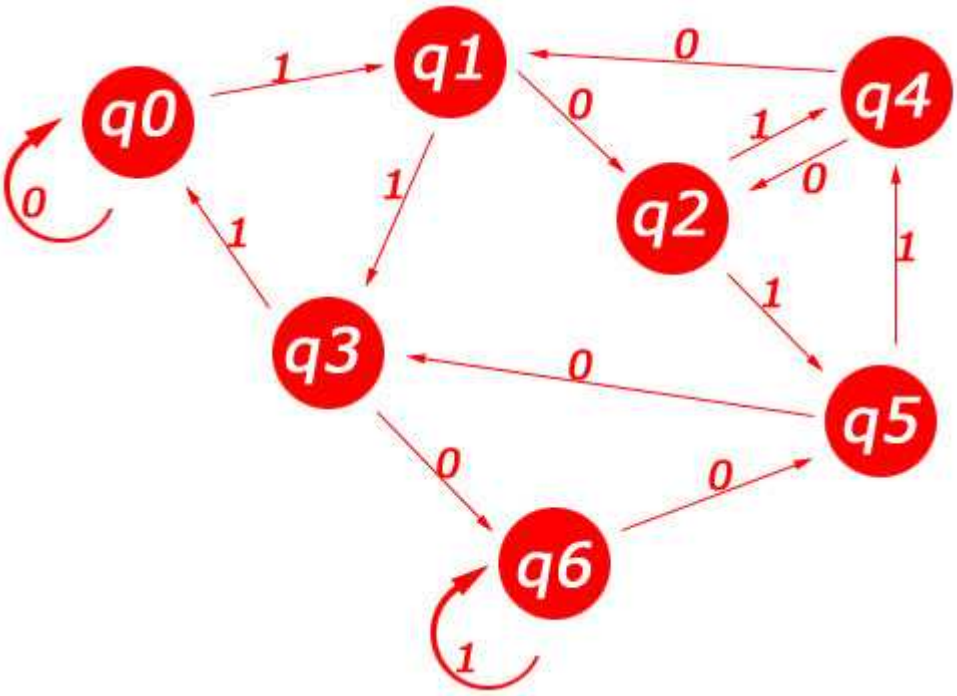
8. Equivalenza tra automa di Moore e automa di Mealy

Teorema: Se M1=(Q, Σ, Δ, δ, λ, q0) è una macchina di Moore, allora esiste una macchina di Mealy equivalente M2= (Q, Σ, Δ, δ, λ', q0)
dove $\lambda'(q, a) = \lambda(\delta(q, a))$ con $a \in \Sigma$

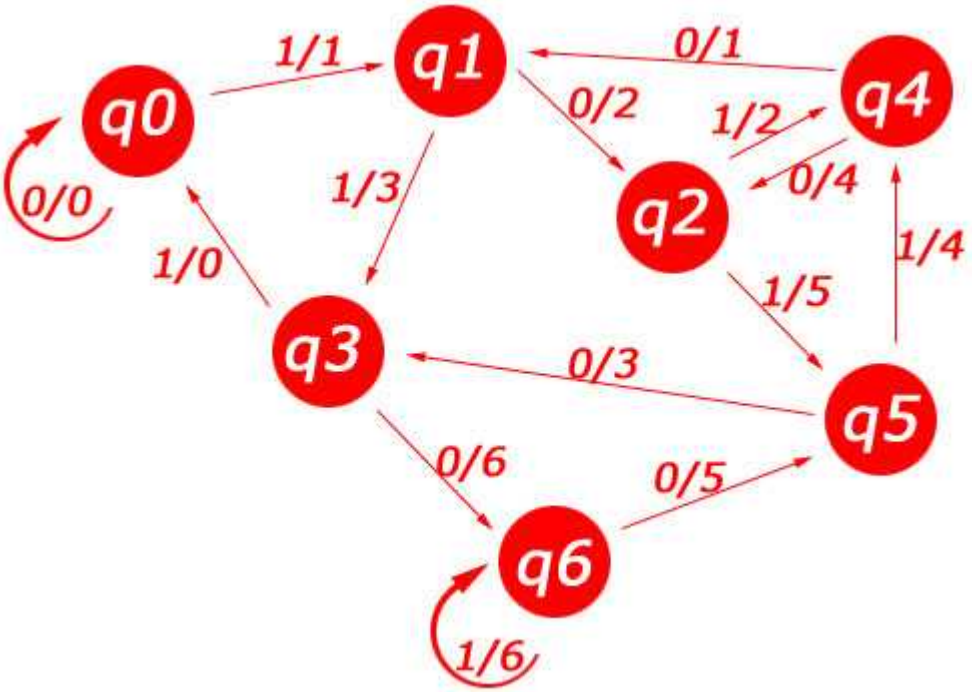
Come si può vedere facilmente dal teorema, nel passaggio dall'automa di Moore a quello di Mealy, viene modificata solamente la funzione di transizione λ e quello che avviene è semplicemente il *"passaggio dell'output dallo stato a cui era associato all'arco che vi punta"*.

Si può vedere anche graficamente:

Automa di Moore



Automa di Mealy



9. Algoritmo in C che trasforma un automa di Moore in un automa di Mealy

```
// L'algoritmo riceve in input le funzioni delta e lambda dell'automa di Moore
// e restituisce in output le funzioni delta' e lambda' dell'automa di Mealy
// secondo la relazione :  lam'(q,s) = lam[ del(q,s) ]

#include <stdio.h>
#define dim 30

int main()
{ int i,j,m,n,x,g;
  char c;
  char s[dim];        // alfabeto di input e di output
  int del[dim][dim]; // funzione delta dell'automa di Moore
  char lam[dim];      // funzione lamda dell'automa di Moore

  printf("\n\n\nAlgoritmo per passare dall'automa di Moore all'automa di
Mealy\n\n\n");
  printf("1. Immettere le informazioni sull'automa di Moore.\n\n");
  printf("Inserire la dimensione dell'alfabeto di input [max %d] : ",dim);
  scanf("%d",&n);
  printf("\nImmettere i simboli dell'alfabeto.\n");
  for(i=0; i<n; i++)
    {  printf("\n\t%d-o simbolo: ",i+1);
       scanf("\n%c",&s[i]);}
  printf("\nInserire il numero degli stati [max %d] : ",dim);
  scanf("%d",&m);
  printf("\n  stato q0 = stato iniziale;\n\n  stati q1..q%d = altri
stati.\n\n",m-1);
  printf("\nImmettere la funzione di transizione Delta.\n");
  for (i=0; i<m; i++)
  for (j=0; j<n; j++)
    { printf("\n\tstato: q%d; simbolo: %c;\t Inserire NUOVO STATO : q",i,s[j]);
      scanf("%d",&del[i][j]); }
  printf("\n\nImmettere la funzione di transizione Lambda.\n");
  for (i=0; i<m; i++)
    { printf("\n\tstato: q%d; output associato: ",i);
      scanf("\n%c",&lam[i]); }


  printf("\n\n\n2. Conferma dei dati memorizzati.\n\n");
  printf("Funzione di transizione Delta per l'automa di Mealy.\n");
  printf("\n\tsimbolo \t");
  for (j=0; j<n; j++) printf("%c\t",s[j]);
  for (i=0; i<m; i++)
    { printf("\n\n\tStato q%d\t",i);
      for (j=0; j<n; j++) printf("q%d\t",del[i][j]);    }
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&g);
  printf("\n\nFunzione di transizione Lambda per l'automa di Mealy.\n");
  printf("\n\t Stato\tOutput");
  for (i=0; i<m; i++) printf("\n\n\t  q%d\t  %c\t",i,lam[i]);
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&g);
  printf("\n\n\n3. Costruzione dell'automa di Mealy.\n\n");
  printf("\nFunzioni LAMBDA per l'automa di Mealy.\n\n");
  printf("\t-simbolo-\t");
  for (j=0; j<n; j++) printf("%c\t",s[j]);
  for (i=0; i<m; i++)
    { printf("\n\n\tstato q%d\t",i);
      for (j=0; j<n; j++) printf("%c\t",lam[del[i][j]]); }
  printf("\n\nProgramma terminato.\n\n"); }
```

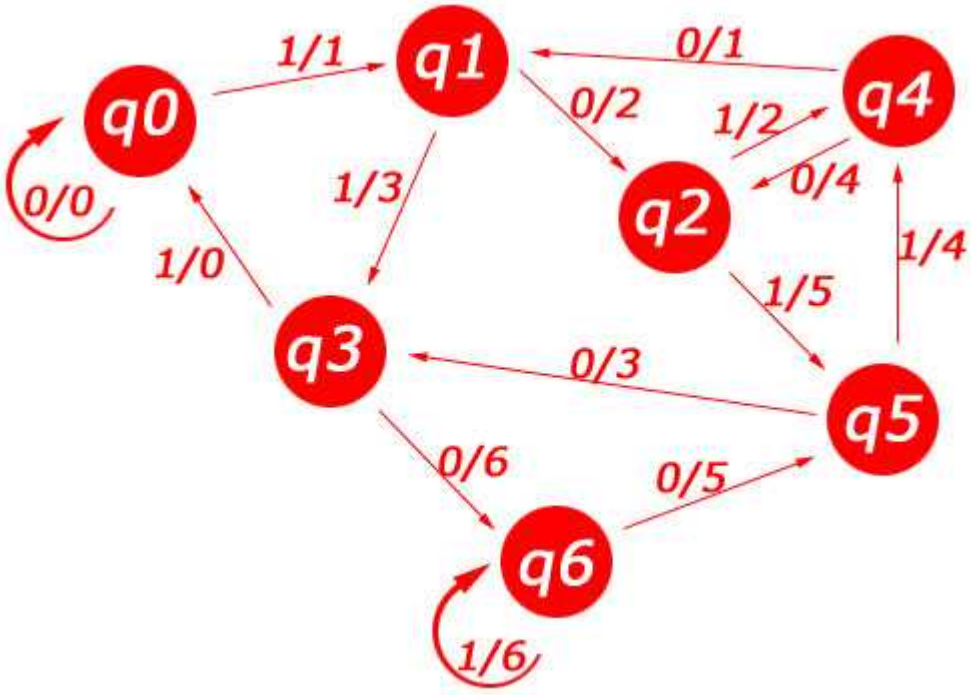
10. Equivalenza tra automa di Mealy e automa di Moore

Teorema: Se M1=(Q, Σ, Δ, δ, λ, q0) è una macchina di Mealy, allora esiste una macchina di Moore equivalente M2= (Q × Δ, Σ, Δ, δ', λ', [q0, b]) con b∈Δ
dove δ'((q, b), a) = (δ(q, a), λ(q, a)) e λ' ((q, b)) = b con b∈Δ e a∈Σ

Nel nostro caso si ha:

Σ = {0, 1}
Δ = {0, 1, 2, 3, 4, 5, 6}
Q = {q0, q1, q2, q3, q4, q5, q6}
Q'= Q × Δ = {(q0, 0), (q0, 1), … , (q0, 6), (q1, 0), … , (q1, 6), …… , (q6, 0), … , (q6, 6)}

L'automa di Mealy di partenza sarà:



Le corrispondenti funzioni λ e δ sono definite come segue:

$\lambda : Q \times \Sigma \longrightarrow \Delta$

λ(q0,0) = 0	λ(q1,0) = 2	λ(q2,0) = 4	λ(q3,0) = 6
λ(q0,1) = 1	λ(q1,1) = 3	λ(q2,1) = 5	λ(q3,1) = 0

λ(q4,0) = 1	λ(q5,0) = 3	λ(q6,0) = 5
λ(q4,1) = 2	λ(q5,1) = 4	λ(q6,1) = 6

$\delta : Q \times \Sigma \longrightarrow Q$

δ(q0,0) = q0	δ(q1,0) = q2	δ(q2,0) = q4	δ(q3,0) = q6
δ(q0,1) = q1	δ(q1,1) = q3	δ(q2,1) = q5	δ(q3,1) = q0

δ(q4,0) = q1	δ(q5,0) = q3	δ(q6,0) = q5
δ(q4,1) = q2	δ(q5,1) = q4	δ(q6,1) = q6

Per passare dall'automa di Mealy a quello di Moore bisogna che sia verificata la seguente relazione:

δ'((q,b),a) = (δ(q,a), λ(q,a)) e λ' ((q,b)) = b

Iniziamo con lo stato **q0:**

δ'((q0,0),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,0)) = 0
δ'((q0,0),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,1),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,1)) = 1
δ'((q0,1),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,2),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,2)) = 2
δ'((q0,2),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,3),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,3)) = 3
δ'((q0,3),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,4),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,4)) = 4
δ'((q0,4),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,5),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,5)) = 5
δ'((q0,5),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

δ'((q0,6),0) = (δ(q0,0), λ(q0,0)) = (q0,0) λ' ((q0,6)) = 6
δ'((q0,6),1) = (δ(q0,1), λ(q0,1)) = (q1,1)

Come si può facilmente notare, per calcolare δ' dello stato (q,b) leggendo a, è ininfluente il valore **b**, come lo è **q** per calcolare λ', così possiamo semplificare notevolmente la costruzione sia di δ' che di λ'

Stato q1:

δ'((q1,0),0) = δ'((q1,1),0) = δ'((q1,2),0) = δ'((q1,3),0) = δ'((q1,4),0) =
= δ'((q1,5),0) = δ'((q1,6),0) = **(δ(q1,0), λ(q1,0)) = (q2,2)**

δ'((q1,0),1) = δ'((q1,1),1) = δ'((q1,2),1) = δ'((q1,3),1) = δ'((q1,4),1) =
= δ'((q1,5),1) = δ'((q1,6),1) = **(δ(q1,1), λ(q1,1)) = (q3,3)**

Stato q2:

δ'((q2,0),0) = δ'((q2,1),0) = δ'((q2,2),0) = δ'((q2,3),0) = δ'((q2,4),0) =
= δ'((q2,5),0) = δ'((q2,6),0) **= (q4,4)**

δ'((q2,0),1) = δ'((q2,1),1) = δ'((q2,2),1) = δ'((q2,3),1) = δ'((q2,4),1) =
= δ'((q2,5),1) = δ'((q2,6),1) = **(q5,5)**

Stato q3:

δ'((q3,0),0) = δ'((q3,1),0) = δ'((q3,2),0) = δ'((q3,3),0) = δ'((q3,4),0) =
= δ'((q3,5),0) = δ'((q3,6),0) **= (q6,6)**

δ'((q3,0),1) = δ'((q3,1),1) = δ'((q3,2),1) = δ'((q3,3),1) = δ'((q3,4),1) =
= δ'((q3,5),1) = δ'((q3,6),1) = **(q0,0)**

Stato q4:

δ'((q4,0),0) = δ'((q4,1),0) = δ'((q4,2),0) = δ'((q4,3),0) = δ'((q4,4),0) =
= δ'((q4,5),0) = δ'((q4,6),0) **= (q1,1)**

δ'((q4,0),1) = δ'((q4,1),1) = δ'((q4,2),1) = δ'((q4,3),1) = δ'((q4,4),1) =
= δ'((q4,5),1) = δ'((q4,6),1) = **(q2,2)**

Stato q5:

δ'((q5,0),0) = δ'((q5,1),0) = δ'((q5,2),0) = δ'((q5,3),0) = δ'((q5,4),0) =
= δ'((q5,5),0) = δ'((q5,6),0) **= (q3,3)**

δ'((q5,0),1) = δ'((q5,1),1) = δ'((q5,2),1) = δ'((q5,3),1) = δ'((q5,4),1) =
= δ'((q5,5),1) = δ'((q5,6),1) = **(q4,4)**

Stato q6:

δ'((q6,0),0) = δ'((q6,1),0) = δ'((q6,2),0) = δ'((q6,3),0) = δ'((q6,4),0) =
= δ'((q6,5),0) = δ'((q6,6),0) **= (q5,5)**

δ'((q6,0),1) = δ'((q6,1),1) = δ'((q6,2),1) = δ'((q6,3),1) = δ'((q6,4),1) =
= δ'((q6,5),1) = δ'((q6,6),1) = **(q6,6)**

Per la funzione λ' (che definisce l'output associato ad ogni stato) si ha che :
per ogni q ∈ Q'= Q x Δ

λ' ((q,0)) = 0
λ' ((q,1)) = 1
λ' ((q,2)) = 2
λ' ((q,3)) = 3
λ' ((q,4)) = 4
λ' ((q,5)) = 5
λ' ((q,6)) = 6

L'automa di Moore così trovato non è disegnabile, in quanto è caratterizzato da un grafo di 49 stati e di 98 archi.

11. Algoritmo in C che trasforma un automa di Mealy in un automa di Moore

```
/* Trasformazione Mealy -> Moore
 L'algoritmo riceve in input le funzioni delta' e lambda' dell'automa di Mealy
 e restituisce in output le funzioni delta e lambda dell'automa di Moore
 secondo le relazioni :      Q = Q'x B, B=alf di output,    quindi   q = [q',b]
                             del([q',b], s) =  [ del(q,s), lam(q,s) ]
                             lam([q',b]) =  b                              */
#include <stdio.h>
#define dim 30

main()
{ int  n,m,i,j,k,x,y,z,q;
  char s[dim];           // alfabeto input
  char out[dim];         // alfabeto output
  int  del[dim][dim];    // funz delta
  char lam[dim][dim];    // funz lamda

   printf("\n\n\nAlgoritmo per passare da un automa di Mealy ad un automa di
Moore.\n\n\n");
  printf("1. Immettere le informazioni sull'automa di Mealy.\n\n");
  printf("Inserire la dimensione dell'alfabeto di input[max %d] : ",dim);
  scanf("%d",&n);
  printf("\nImmettere i simboli dell'alfabeto.\n");
  for(i=0; i<n; i++)
    {  x=i+1; printf("\n\t%d-o simbolo: ",x);
       scanf("\n%c",&s[i]); }
  printf("\nInserire il numero degli stati [max %d]: ",dim);
  scanf("%d",&m);
  printf("\n  stato q0 = stato iniziale;\n\n");
  printf("stati q1..q%d = altri stati.\n\n",m-1);
  printf("\nImmettere la funzione di transizione Delta.\n");
  for (i=0; i<m; i++)
  for (j=0; j<n; j++)
    { printf("\n\tstato: q%d; simbolo: %c;\t",i,s[i]);
      printf(" -> Inserire NUOVO STATO: q");
      scanf("%d",&del[i][j]); }
```

```
  printf("\n\nInserire la dimensione dell'alfabeto di output (max %d): ",dim);
  scanf("%d",&z);
  printf("\nImmettere gli output.\n");
  for (i=0; i<z; i++)
    { printf("\n\t%d-o output: ",i+1);
      scanf("\n%c",&out[i]);}
  printf("\n\nImmettere la funzione di transizione Lambda.\n");
  for (i=0; i<m; i++)
  for (j=0; j<n; j++)
    { printf("\n\tstato: q%d; simbolo: %c;\t -> output associato: ",i,s[j]);
      scanf("\n%c",&lam[i][j]);    }

  printf("\n\n\n2. Conferma dei dati memorizzati.\n\n");
  printf("Funzione di transizione Delta per l'automa di Mealy.\n");
  printf("\n\tsimbolo \t");
  for (j=0; j<n; j++) printf("%c\t",s[j]);
  for (i=0; i<m; i++)
    { printf("\n\n\tStato q%d\t",i);
      for (j=0; j<n; j++) printf("q%d\t",del[i][j]);    }
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&q);
  printf("\n\nFunzione di transizione Lambda per l'automa di Mealy.\n");
  printf("\n\tsimbolo \t");
  for (j=0; j<n; j++) printf("%c\t",s[j]);
  for (i=0; i<m; i++)
    { printf("\n\n\tStato q%d\t",i);
      for (j=0; j<n; j++) printf("%c\t",lam[i][j]); }
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&q);

  printf("\n\n\n3. Costruzione dell'automa di Moore\n\n");
  printf("L'automa di Moore associato (non minimizzato) avrà %d stati.\n",m*z);
  printf("\n\nInsieme dei nuovi stati.\n\n\t");
  for (i=0; i<m; i++)
    { for (j=0; j<z; j++) printf("[q%d/%c]\t",i,out[j]);
      printf("\n\n\t"); }
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&q);
  printf("\nFunzioni DELTA per l'automa di Moore\n\n");
  printf("\t---simbolo--\t\t");
  for (k=0; k<n; k++) printf("%c\t",s[k]);
  for (i=0; i<m; i++)
  for (j=0; j<z; j++)
    { printf("\n\n\tstato: [q%d/%c]\t\t",i,out[j]);
      for (k=0; k<n; k++) printf("[q%d/%c]\t",del[i][k],lam[i][k]);  }
  printf("\n\nDigitare 0 e premere INVIO per continuare... ");
  scanf("\n%c",&q);
  printf("\nFunzioni LAMBDA per l'automa di Moore.\n\n");
  printf("\t---simbolo--\t\t");
  for (k=0; k<n; k++) printf("%c\t",s[k]);
  for (i=0; i<m; i++)
  for (j=0; j<z; j++)
    { printf("\n\n\tstato: [q%d/%c]\t\t",i,out[j]);
      for (k=0; k<n; k++) printf("%c\t",out[j]);  }
  printf("\n\nProgramma terminato.\n\n");
}
```

Quindi riassumendo l'automa di Moore verrà costruito nel seguente modo:

Insieme degli stati :

$$Q' = \left\{ \begin{array}{cccc} [q0/0] & [q0/1] & \dots & [q0/6] \\ [q1/0] & [q1/1] & \dots & [q1/6] \\ \dots & \dots & \dots & \dots \\ [q6/0] & [q6/1] & \dots & [q6/6] \end{array} \right\}$$

Funzione di transizione delta :

$$\delta'([q, b], a) = [\delta'(q, a), \lambda'(q, a)] \qquad \forall q \in Q,\ \forall a \in \Sigma,\ \forall b \in \Delta$$

δ'	0		1		2		3		4		5		6	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1
q0	[q0/0]	[q1/1]	[q0/0]	[q1/1]	[q0/0]	[q1/1]	[q0/0]	[q1/1]	[q0/0]	[q1/1]	[q0/0]	[q1/1]	[q0/0]	[q1/1]
q1	[q2/2]	[q3/3]	[q2/2]	[q3/3]	[q2/2]	[q3/3]	[q2/2]	[q3/3]	[q2/2]	[q3/3]	[q2/2]	[q3/3]	[q2/2]	[q3/3]
q2	[q4/4]	[q5/5]	[q4/4]	[q5/5]	[q4/4]	[q5/5]	[q4/4]	[q5/5]	[q4/4]	[q5/5]	[q4/4]	[q5/5]	[q4/4]	[q5/5]
q3	[q6/6]	[q0/0]	[q6/6]	[q0/0]	[q6/6]	[q0/0]	[q6/6]	[q0/0]	[q6/6]	[q0/0]	[q6/6]	[q0/0]	[q6/6]	[q0/0]
q4	[q1/1]	[q2/2]	[q1/1]	[q2/2]	[q1/1]	[q2/2]	[q1/1]	[q2/2]	[q1/1]	[q2/2]	[q1/1]	[q2/2]	[q1/1]	[q2/2]
q5	[q3/3]	[q4/4]	[q3/3]	[q4/4]	[q3/3]	[q4/4]	[q3/3]	[q4/4]	[q3/3]	[q4/4]	[q3/3]	[q4/4]	[q3/3]	[q4/4]
q6	[q5/5]	[q6/6]	[q5/5]	[q6/6]	[q5/5]	[q6/6]	[q5/5]	[q6/6]	[q5/5]	[q6/6]	[q5/5]	[q6/6]	[q5/5]	[q6/6]

Funzione di transizione lamda :

$$\lambda'([q, b], a) = b \qquad \forall q \in Q,\ \forall a \in \Sigma,\ \forall b \in \Delta$$

λ'	0	1	2	3	4	5	6
q0	0	1	2	3	4	5	6
q1	0	1	2	3	4	5	6
q2	0	1	2	3	4	5	6
q3	0	1	2	3	4	5	6
q4	0	1	2	3	4	5	6
q5	0	1	2	3	4	5	6
q6	0	1	2	3	4	5	6