Una TM e’ una 7-pla (Q,Σ,Γ,δ,q 0 ,B,F) con:

Un insieme finito di stati Q

Un alfabeto di input Σ

Un alfabeto del nastro Γ (contiene Σ)

Una funzione di transizione δ

Uno stato iniziale q 0 ∈Q

Un simbolo blank B∈Γ-Σ

Tutto il nastro, meno l’input, all’inizio e’ blank

Un insieme di stati finali F⊆Q

----------------------

δ(q, Z) e’ indefinito oppure e’ una tripla dalla forma (p, Y, D).

p e’ uno stato

Y e’ un nuovo simbolo del nastro

D e’ una direzione, L o R

-------------------  
  
Descrizione Istantantea (ID)

αqβ

nastro – testina / stato – carattere che legge la testina – nastro

q00⊦0q0⊦00q⊦0q01⊦00q1⊦000f

-----------------------------  
  
Linguaggio di una TM:

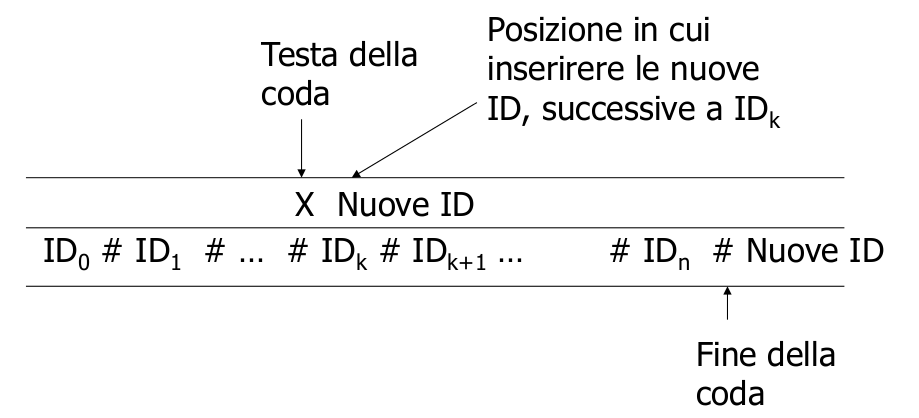
L(M) = {w | q 0 w⊦\*αqβ, con q finale}

-------------------------------

**Tm NON Deterministiche:**

La funzione di transizione ha, come output, un insieme di triple stato-simbolo-direzione (le TM deterministiche hanno al piu' una tripla)

**Simulare una NTM con una DTM:**

****

Ci sono 2 tracce:

La prima traccia contiene le ID della NTM, che vengono lette dalla DTM.

La DTM, quindi, cerca nell'ID corrente lo stato della ID per poter vedere quali sono le possibili mosse.

Una seconda traccia è usata per marcare la ID corrente (X) e per creare le nuove ID in caso di non determinismo, ne crea una per ogni mossa possibile.

Tutte le ID create poi vengono aggiunte nella prima traccia in fondo alla coda.

Dopodichè, sposa la marcatura avanti e prende la prossima ID.

Si ferma quando trova un ID con stato accettante.

**PROPRIETÀ DI CHIUSURA PER LINGUAGGI RICORSIVI E R.E.**

Entrambi chiusi rispetto a unione, concatenazione, stella di Kleene, reverse

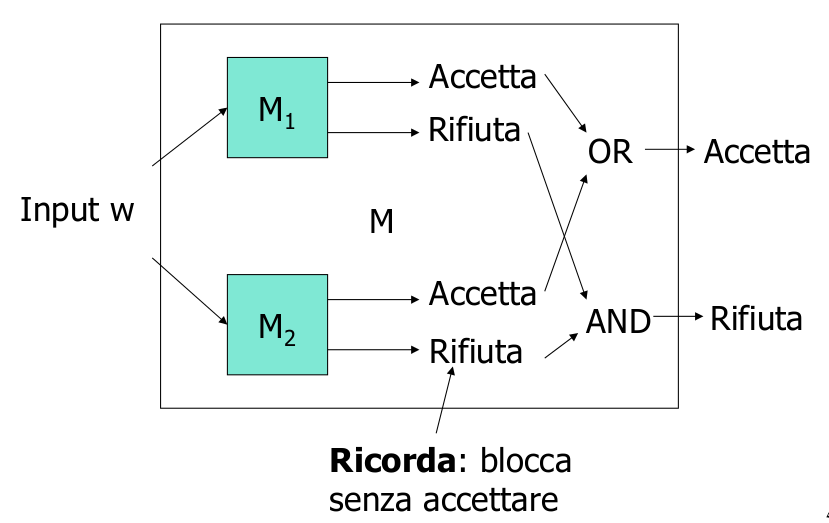
e intersezione

Ricorsivi chiusi anche rispetto a differenza e complemento

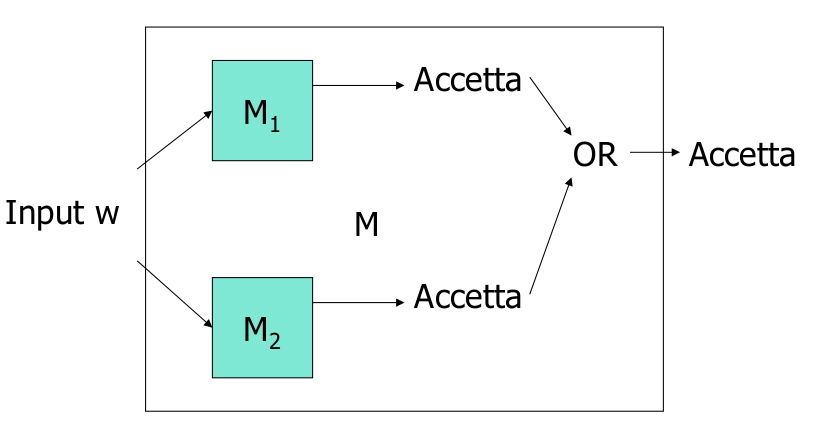
**Unione:**

Costruiamo una TM con 2 nastri che prima copia l’input sul secondo nastro e poi simula “in parallelo” le due TM (M1 sul primo nastro, M2 sul secondo nastro)

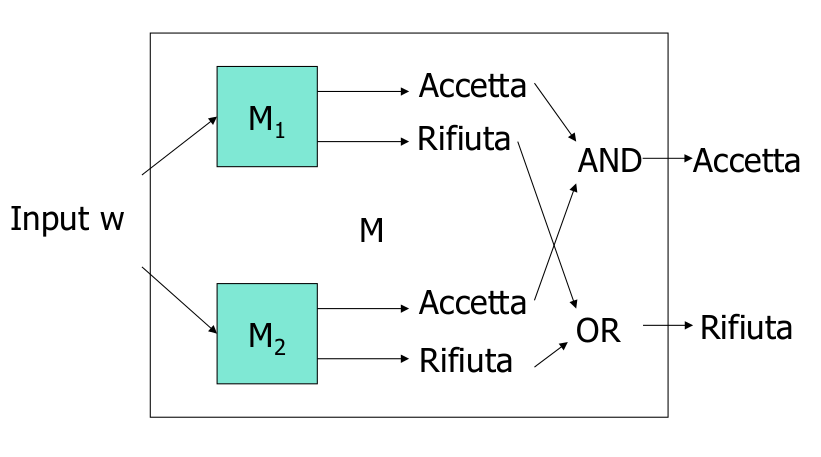
**Unione - Ricorsivi**



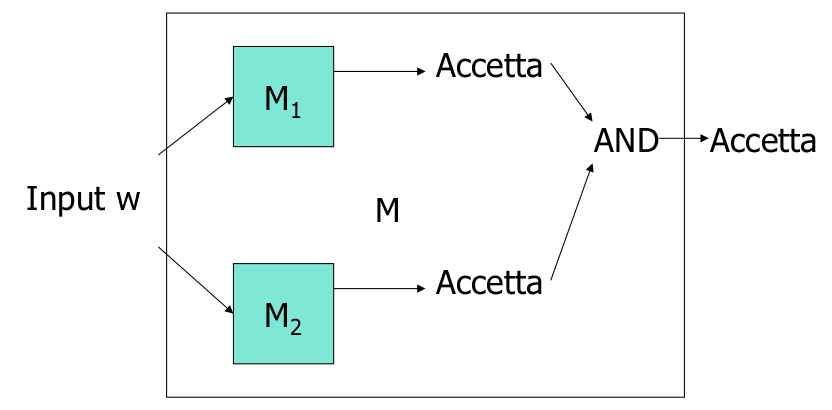
**Unione: RE**



**Intersezione – Ricorsivi:**

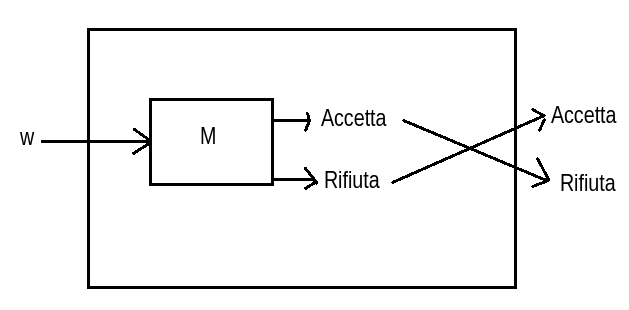


**Intersezione – R.E.**



**Differenza – Complemento - Ricorsivi:**

Sempre come la figura sopra, M1 accetta e M2 non accetta (perché è il complemento)

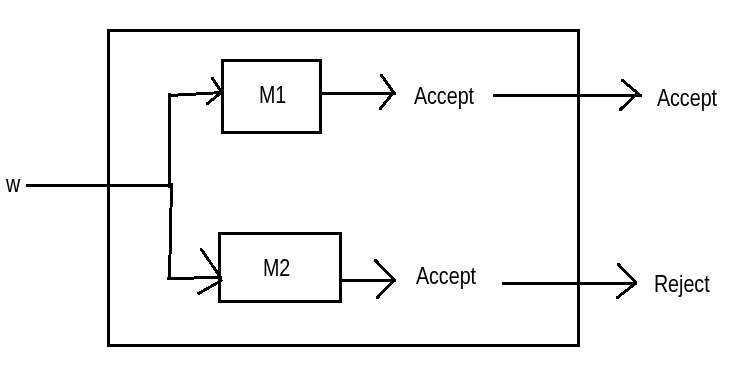


TM per il complemento di L.

Se L è ricorsivo, allora anche L complementato è ricorsivo, perché l'algoritmo termina sempre.

**Differenza – Complemento – RE:**

Se L e L complementato sono R.E., allora sia L che L complementato sono ricorsivi.



TM che accetta sia L che L complementato, che sono R.E.

Se w € L, allora M1 accetta, M2 andrà avanti all'infinito, ma M accetta perché M1 ha accettato.

Se w € L complementato, allora M2 accetta, quindi M rifiuterà.

Per ogni input quindi M o accetta o termina, quindi sia L che L complementato sono ricorsivi.

**Concatenazione – R.E. :**

Siano L 1 =L(M 1 ) e L 2 =L(M 2 )

Assumiamo che M 1 e M 2 siano TM con un solo nastro semi-infinito

Costruiamo una TM a due nastri nondeterministica M:

Costruiamo una TM a due nastri nondeterministica M:

* Prova un troncamento dell’input w = xy (crea un'istanza per ogni troncamento possibile)
* Muove y sul secondo nastro
* Simula M1 su x, M2 su y
* Accetta se entrambe accettano

**Concatenazione – Ricorsivi:**

Come prima, la NTM accetta se entrambe le TM accettano per un qualche troncamento di w = xy, o rifiuta se tutti i possibili troncamenti falliscono.

**Stella di Kleene:**

Come per la concatenzazione, si prova un possibile troncamento e si accetta se vengono accettati tutti i pezzi.