

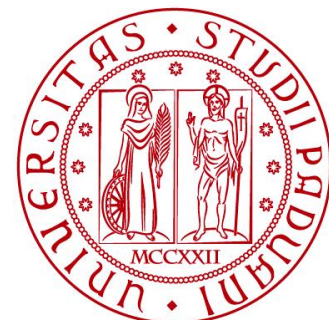
Automi e Linguaggi Formali

**Macchine di Turing, indecidibilità e
problemi intrattabili**

Lamberto Ballan

lamberto.ballan@unipd.it

Padova, 9 Maggio 2017



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Presentazione: come contattarmi / incontrarmi

Docente: Lamberto Ballan

E-mail: lamberto.ballan@unipd.it

Ufficio: Stanza 322, III piano, corridoio CD (Torre Archimede)

Ricevimento: venerdì 15:30-17:30 o su appuntamento

Presentazione: adesso e “futuro prossimo”

Presa di servizio “ufficiale”: Ottobre 2017

Corsi A.A.2017/18: Tecnologie Web (LT), Cognitive Services (LM)

Attualmente: Università di Firenze / Stanford University

EU Marie Curie Fellow (10/2014-09/2017)

Progetto: *Exploiting semantic and social knowledge for visual recognition*

Area di ricerca: AI, Computer Vision, Multimedia, Big (visual) Data



Calendario dell'ultima parte del corso

■ Parte 4: indecidibilità e intrattabilità

- macchine di Turing
- concetto di indecidibilità
- problemi intrattabili
- classi P e NP



Capitoli 8,9,10

Periodo: 9/5/2017 - 9/6/2017

~~Lun: 13:30-15:30~~

Mar: 11:30-12:45, 13:30-15:30

Venerdì: 13:30-15:30

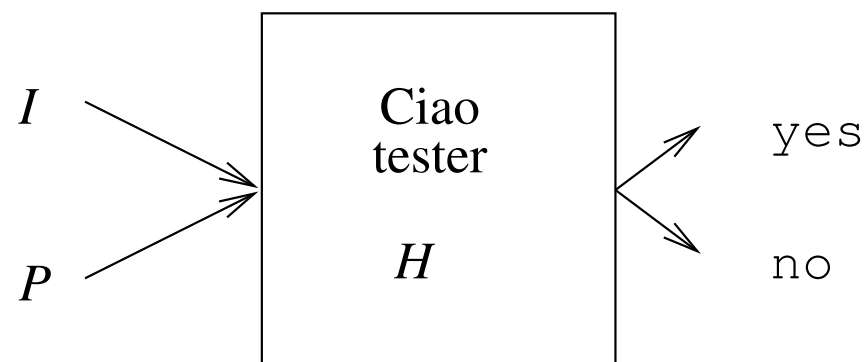
← --- da verificare "in corsa"

Problemi che i calcolatori non possono risolvere

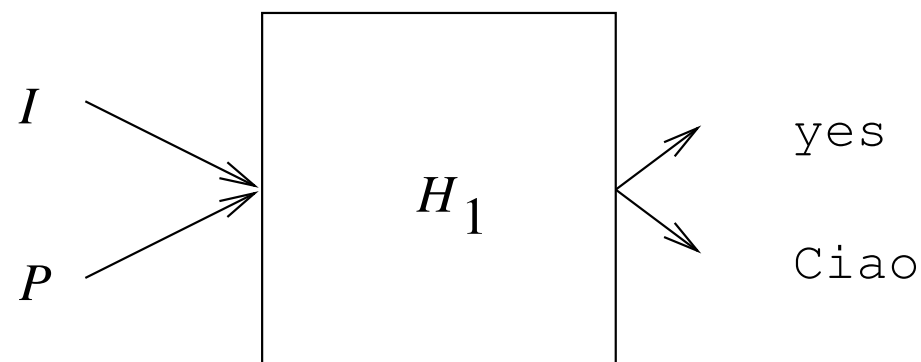
- È importante sapere se un programma è corretto, cioè fa quello che ci aspettiamo
- È facile vedere che il programma contente solo il comando `printf('Ciao')` stampa Ciao
- Ma il programma in Fig. 8.2 del libro?
- Stampa Ciao, dato un input n se e solo se l'equazione $x^n + y^n = z^n$ ha una soluzione dove x , y e z sono interi
- Sappiamo ora (ultimo teorema di Fermat) che stampa Ciao, con l'input $n = 2$, e cicla per sempre su input $n > 2$
- Ci sono voluti 300 anni per provarlo
- Possiamo sperare di avere un programma che prova la correttezza di programmi?

Un ipotetico verificatore H che testa “Ciao mondo”

Supponiamo che esista un programma H tale che, dato un qualunque programma P e un input I , dice yes se P con input I stampa ‘Ciao’, altrimenti dice no

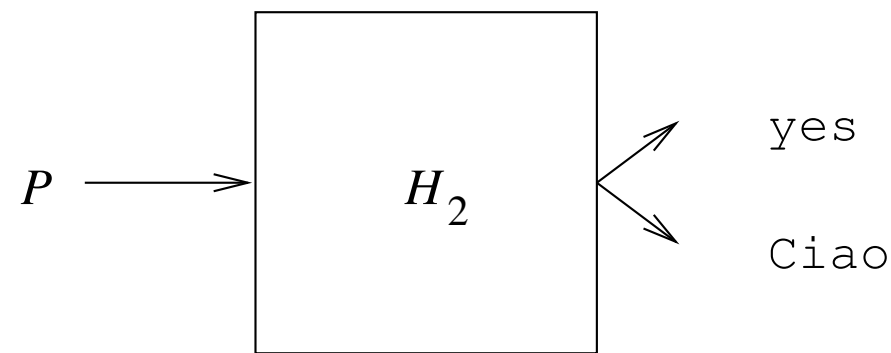


Modifichiamo il comando di stampa di no di H in Ciao.
Otteniamo il programma H_1

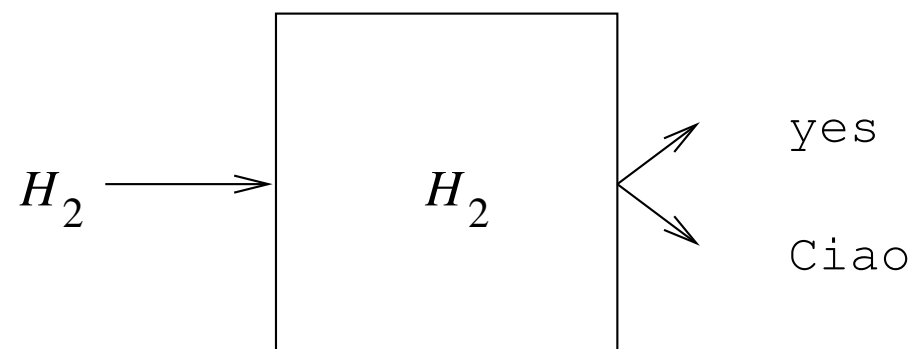


Un ipotetico verificatore H che testa “Ciao mondo”

Modifichiamo H_1 in modo che l'input I sia lo stesso P . Otteniamo il programma H_2 :

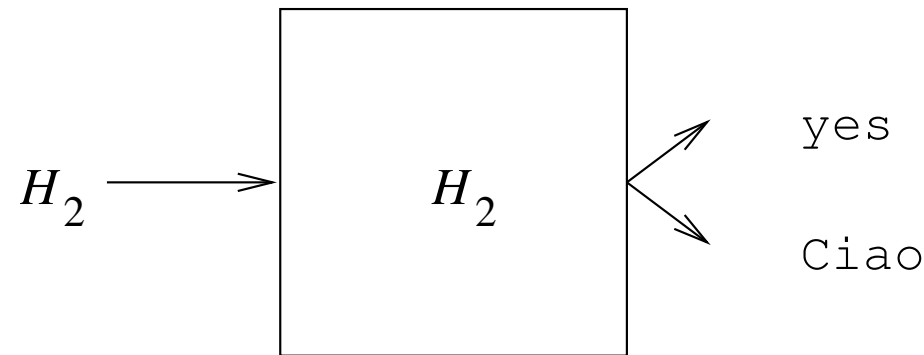


H_2 stampa yes se P con input P stampa Ciao, altrimenti stampa Ciao. Diamo H_2 in input ad H_2 :



Un ipotetico verificatore H che testa “Ciao mondo”

Diamo H_2 in input ad H_2 :



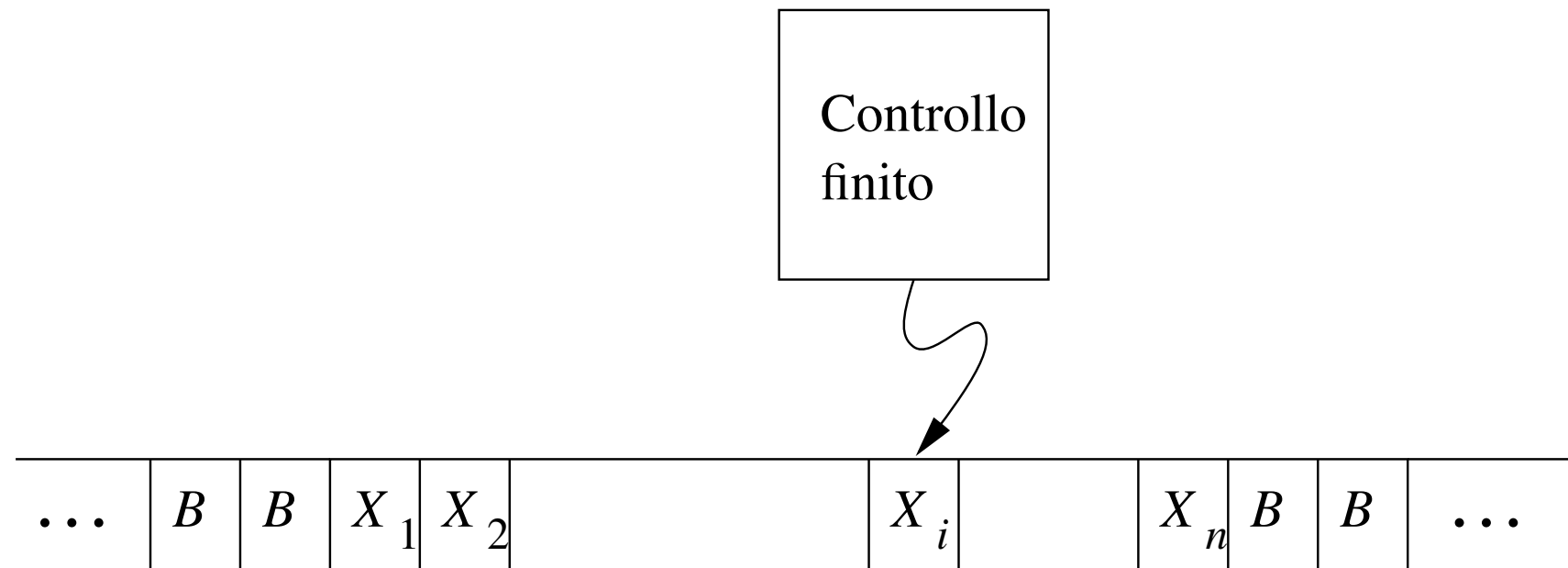
- Se H_2 stampa yes, avrebbe dovuto stampare Ciao
- Se H_2 stampa Ciao, avrebbe dovuto stampare yes
- Quindi H_2 non può esistere
- Quindi neanche H può esistere

Quindi il problema affrontato é **indecidibile**: cioè non esiste nessun programma che, dato un programma qualsiasi ed un input, sappia dire se quel programma stamperà “Ciao mondo”

Problemi indecidibili

- Problemi per cui non esiste alcun programma che li possa risolvere
- Problema: appartenenza di una stringa ad un linguaggio
- Il numero di linguaggi diversi su un alfabeto non è numerabile
- I programmi (stringhe finite su un alfabeto) sono numerabili: si possono ordinare per lunghezza, e poi lessicograficamente \Rightarrow primo programma, secondo programma, ecc.
- Quindi esistono infinitamente più linguaggi che programmi
- Quindi devono esistere problemi indecidibili (Godel 1931)

La macchina di Turing (1937)

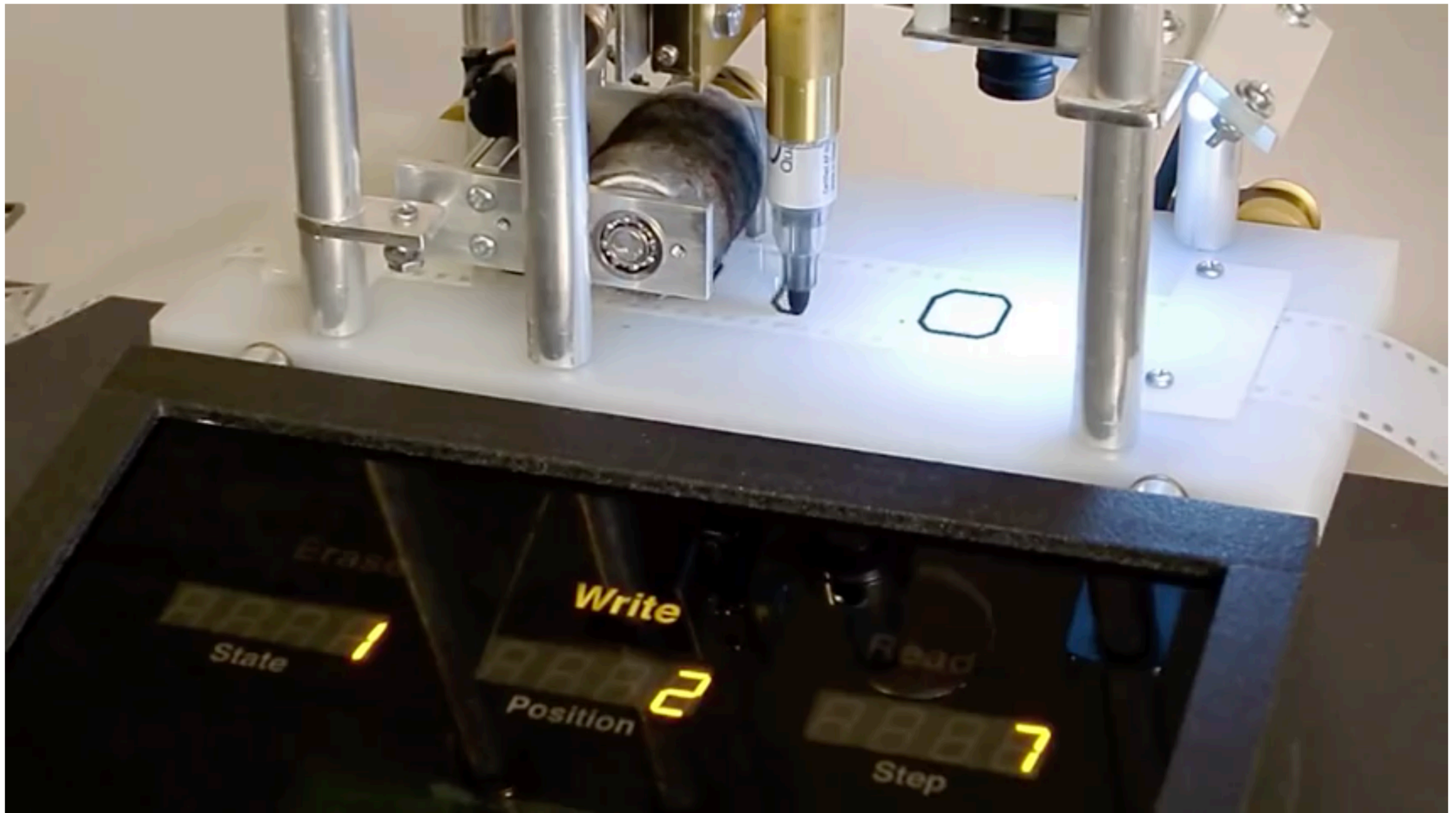


Una macchina di Turing esegue una **mossa** in funzione del suo stato e del simbolo che si trova nella cella visitata dalla testina di lettura del nastro.

In una mossa, la macchina di Turing (TM) compie tre azioni:

- Cambia stato
- Scrive un simbolo di nastro nella cella che visita
- Muove la testina verso sinistra o verso destra

La macchina di Turing (1937)



Notazione formale di una TM come automa

Una macchina di Turing deterministica è una 7-tupla

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F),$$

dove

- 1 Q è un insieme finito di stati,
- 2 Σ è un insieme finito di simboli di input,
- 3 Γ è un insieme finito di simboli di nastro,
- 4 δ è una funzione di transizione da Q a $Q \times \{L, R\}$,
- 5 q_0 è lo stato iniziale,
- 6 $B \in \Gamma$ è il simbolo blank, e
- 7 $F \subseteq Q$ è l'insieme di stati finali

Descrizione istantanee di una TM

Una TM cambia configurazione dopo ogni mossa.
Usiamo le descrizioni istantanee (ID) per descrivere le configurazioni.

Una ID è una stringa della forma

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n$$

dove

- 1** q è lo stato della TM
- 2** $X_1 X_2 \cdots X_n$ è la porzione non-blank del nastro
- 3** La testina è sopra il simbolo i -esimo

Le mosse e il linguaggio di una TM

Useremo \vdash_M per indicare una mossa di M da una configurazione ad un'altra.

- Supponiamo $\delta(q, X_i) = (p, Y, L)$. Allora

$$X_1 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \vdash_M X_1 \cdots X_{i-2} p X_{i-1} Y X_{i+1} \cdots X_n$$

- Se $\delta(q, X_i) = (p, Y, R)$. Allora

$$X_1 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \vdash_M X_1 \cdots X_{i-1} Y p X_{i+1} \cdots X_n$$

Indichiamo la chiusura riflessiva e transitiva di \vdash_M con \vdash_M^* .

Diagrammi di transizione per le TM

- Possiamo rappresentare le transizioni di una TM graficamente come abbiamo fatto per i PDA
- Il *diagramma di transizione* consiste in un insieme di nodi corrispondenti agli stati della TM
- Ad esempio, data una mossa $\delta(q, X_i) = (p, Y, L)$, avremo:
 - un nodo per ciascun stato q e p
 - un arco che connette q e p con etichetta X_i / YL

Definiamo una TM che accetta $\{0^n 1^n : n \geq 1\}$

- Alla macchina viene data sul nastro una sequenza finita di 0 e 1, preceduta e seguita da B
- La TM cambia in modo alternato uno 0 in X e un 1 in Y finché tutti gli 0 e gli 1 sono “abbinati”
- Cioè, partendo dall'estremità sinistra dell'input, la TM:
 - cambia uno 0 in X e si muove a destra su tutti gli 0 e Y che vede finché arriva a un 1
 - cambia l'1 in Y e si muove a sinistra su tutti gli Y e 0 che vede finché trova una X
 - cerca uno 0 immediatamente a destra e se ne trova uno lo cambia in X e ripete il processo

Definiamo una TM che accetta $\{0^n 1^n : n \geq 1\}$

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$$

dove δ è data dalla seguente tabella

	0	1	X	Y	B
$\rightarrow q_0$	(q_1, X, R)			(q_3, Y, R)	
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
$*q_4$					

Definiamo una TM che accetta $\{0^n 1^n : n \geq 1\}$

Possiamo rappresentare M con il seguente **diagramma di transizione**

