

Informatica per le discipline umanistiche

Lezione 3 – Macchina di Turing

`cristiano.longo@unict.it`



La Macchina Analitica

1837 Charles Babbage progetta la sua *macchina analitica*, capace di effettuare calcoli generali sotto il pieno controllo automatico.



La Macchina Analitica

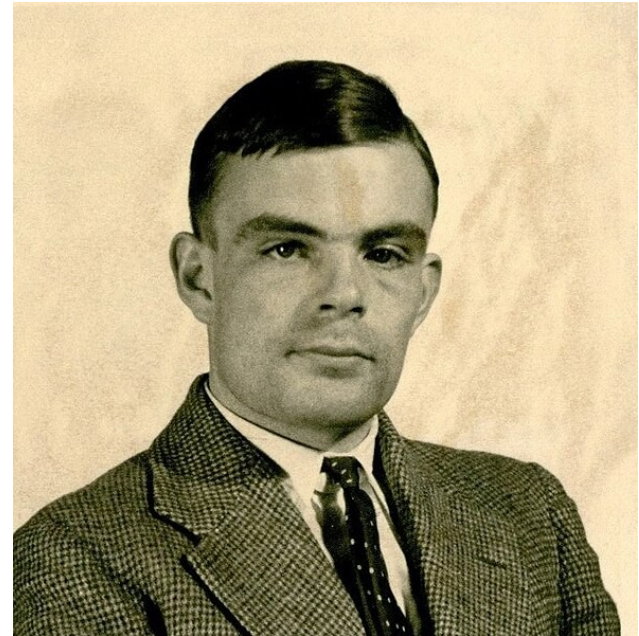
1837 Charles Babbage progetta la sua *macchina analitica*, capace di effettuare calcoli generali sotto il pieno controllo automatico.

Ada (Byron contessa di) Lovelace scrisse svariati programmi per la macchina analitica. È considerata la prima programmatrice della storia.



Gli Algoritmi secondo Alan Turing

Per affrontare il problema della decisione, nel 1936 **Alan Turing** in *On computable Numbers, with an application to the Entscheidungsproblem* presenta la sua *macchina*.



Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una somma.

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
					7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
			2	1	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
		1	6	1	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
				7		

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
		2	1	7		

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
	1	6	1	7		

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: moltiplicazione delle singole cifre

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
					7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
				8	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
			7	8	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
	1	5	7	8	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
1	2	5	7	8	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Gli Algoritmi secondo Turing - intuizione

Turing parte dall'osservazione delle azioni eseguite da una persona impegnata in un calcolo matematico, ad esempio una moltiplicazione. Essa ha a disposizione un foglio a quadretti e un lapis.

		4	2	3	1	x
				7	7	=
	2	9	6	1	7	
2	9	6	1	7		
3	2	5	7	8	7	

Pone la sua attenzione solo su alcuni simboli alla volta.

Scrive simboli in base a quelli su cui ha posto attenzione ed al suo *stato mentale*.

Stato Mentale: somma dei risultati

Macchine di Turing

Una macchina di Turing è costituita

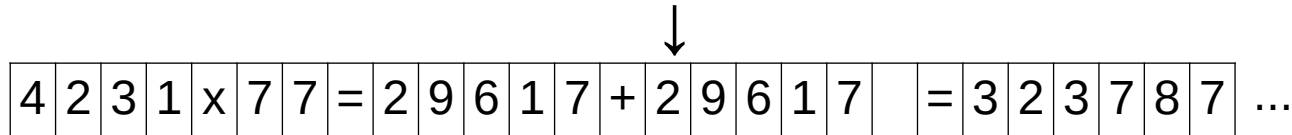
- da un **nastro** infinito;

4	2	3	1	x	7	7	=	2	9	6	1	7	+	2	9	6	1	7		=	3	2	3	7	8	7	...
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	-----

Macchine di Turing

Una macchina di Turing è costituita

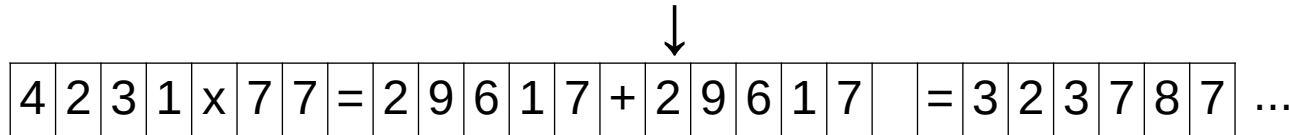
- da un **nastro** infinito;
- una **testina** posizionata in ogni momento su una casella del nastro, in grado di leggere il simbolo nella casella, scrivere un nuovo simbolo e spostarsi a destra o sinistra di una posizione;



Macchine di Turing

Una macchina di Turing è costituita

- da un **nastro** infinito;
- una **testina** posizionata in ogni momento su una casella del nastro, in grado di leggere il simbolo nella casella, scrivere un nuovo simbolo e spostarsi a destra o sinistra di una posizione;
- uno **stato**;



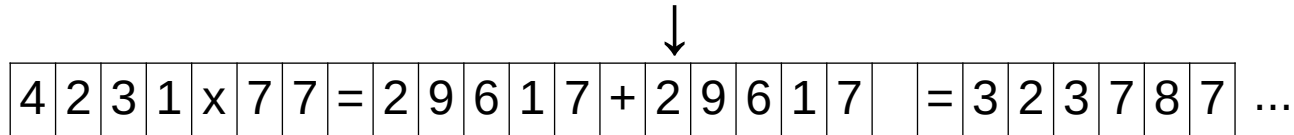
Stato: SOMMA

Macchine di Turing

Una macchina di Turing è costituita

- da un **nastro** infinito;
- una **testina** posizionata in ogni momento su una casella del nastro, in grado di leggere il simbolo nella casella, scrivere un nuovo simbolo e spostarsi a destra o sinistra di una posizione;
- uno **stato**;
- Un insieme finito di **regole di transizione**

(stato, simbolo letto) → (nuovo stato, simbolo da scrivere, spostamento)



Stato: SOMMA

Macchine di Turing – definizione formale

Una macchina di Turing è definita da

- da un insieme finito di stati Q che comprende lo stato iniziale θ ;
- un alfabeto dei simboli Γ che comprende $-$ (blank);
- un insieme finito di **regole di transizione**
(stato, simbolo letto) \rightarrow (nuovo stato, simbolo da scrivere, spostamento $\{<, >, -\}$).

Macchine di Turing – definizione formale

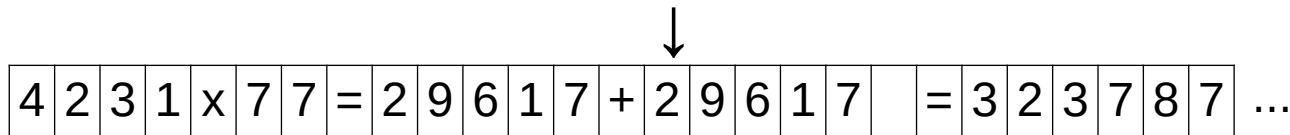
Una macchina di Turing è definita da

- da un insieme finito di stati Q che comprende lo stato iniziale θ ;
- un alfabeto dei simboli Γ che comprende - (blank);
- un insieme finito di regole di transizione

(stato, simbolo letto) \rightarrow (nuovo stato, simbolo da scrivere, spostamento {<, >, -}).

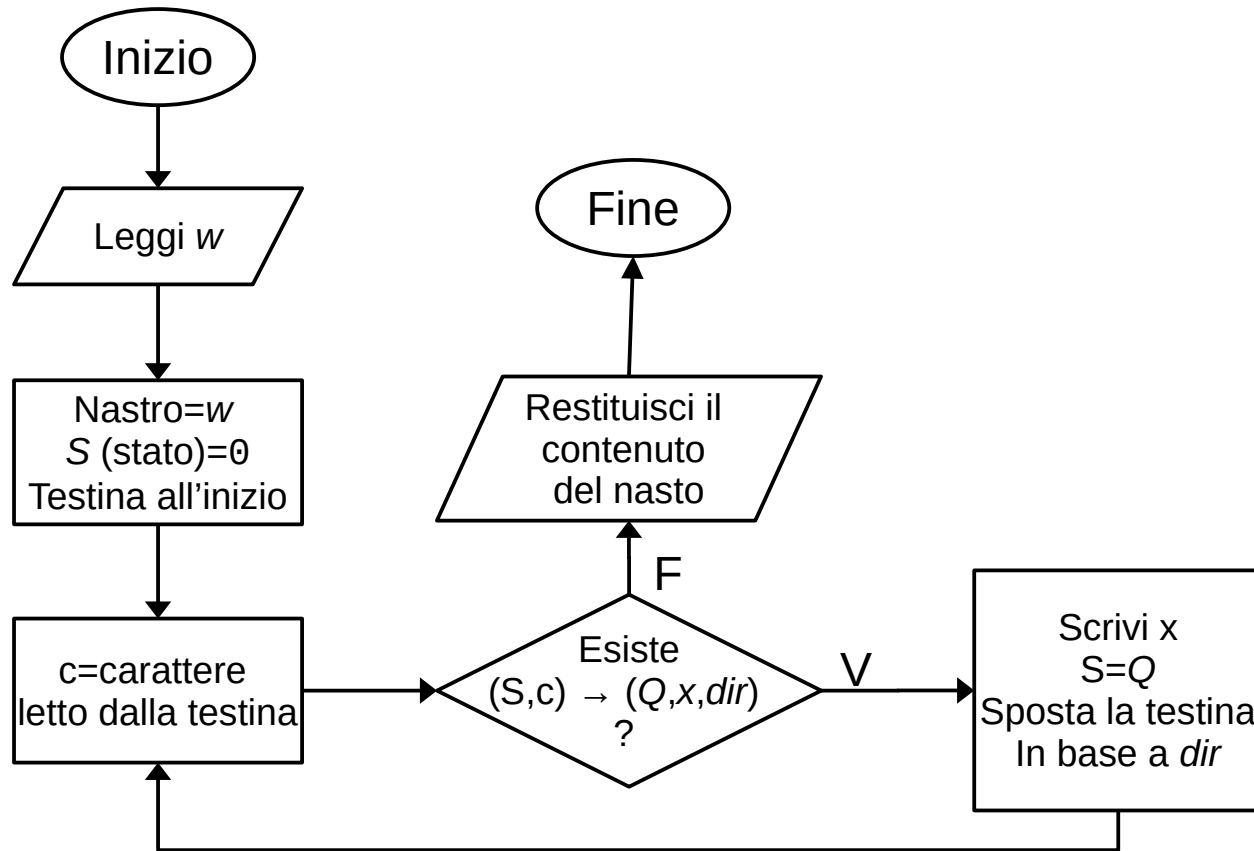
Una **configurazione** è caratterizzata da

- dal contenuto del nastro (omettendo gli infiniti blank nella parte terminale del nastro),
- dalla posizione della testina, e
- dallo stato corrente.



Stato: SOMMA

Macchine di Turing – esecuzione



Macchine di Turing – esempio 1

Problema: data una *stringa* di caratteri X e Y (sequenza finita di X e Y), sostituire Y ad ogni X.

Esempi

Dati di ingresso	Valore restituito
XYYXY	YYYYY
YYY	YYY
XXXX	YYYY

Macchina di Turing

$Q = \{0\}$

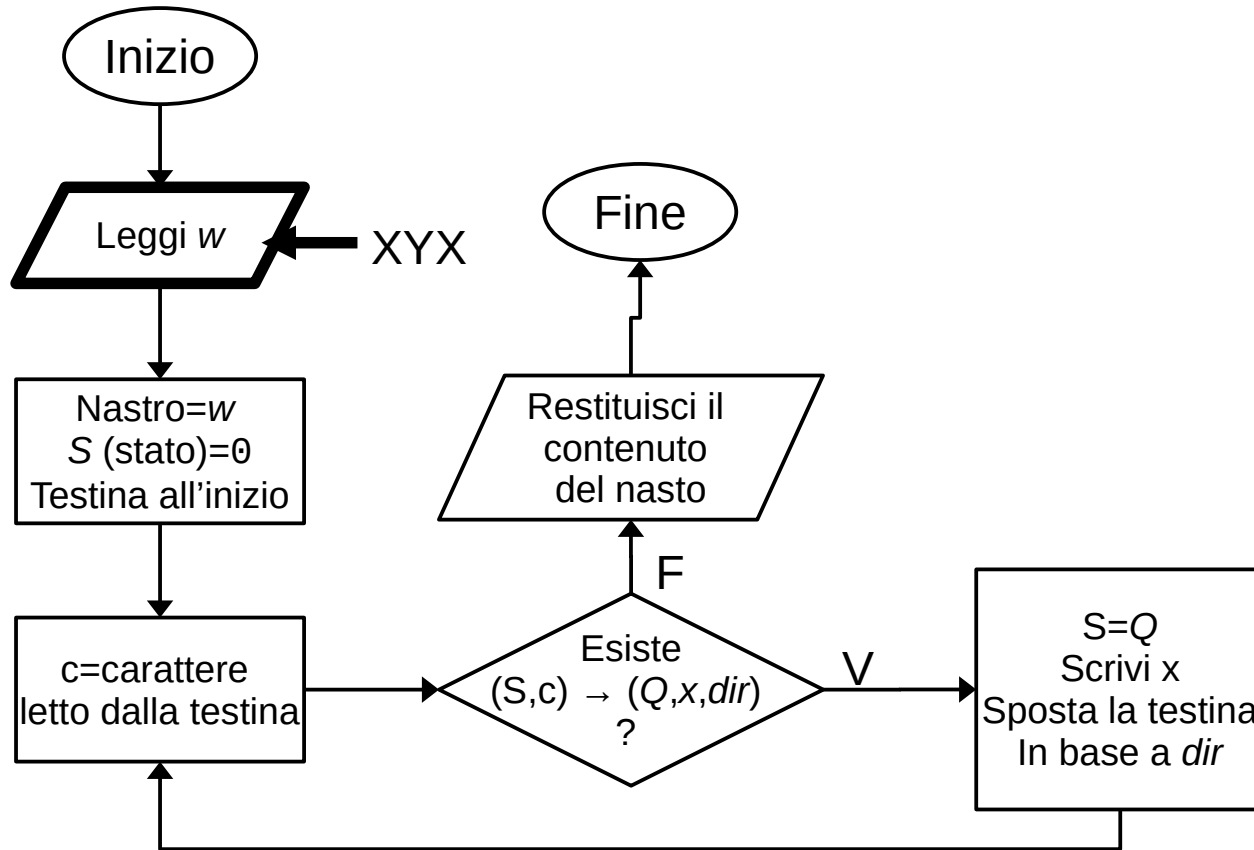
$\Gamma = \{X, Y, -\}$

Regole di transizione

$(0, X) \rightarrow (0, Y, >)$

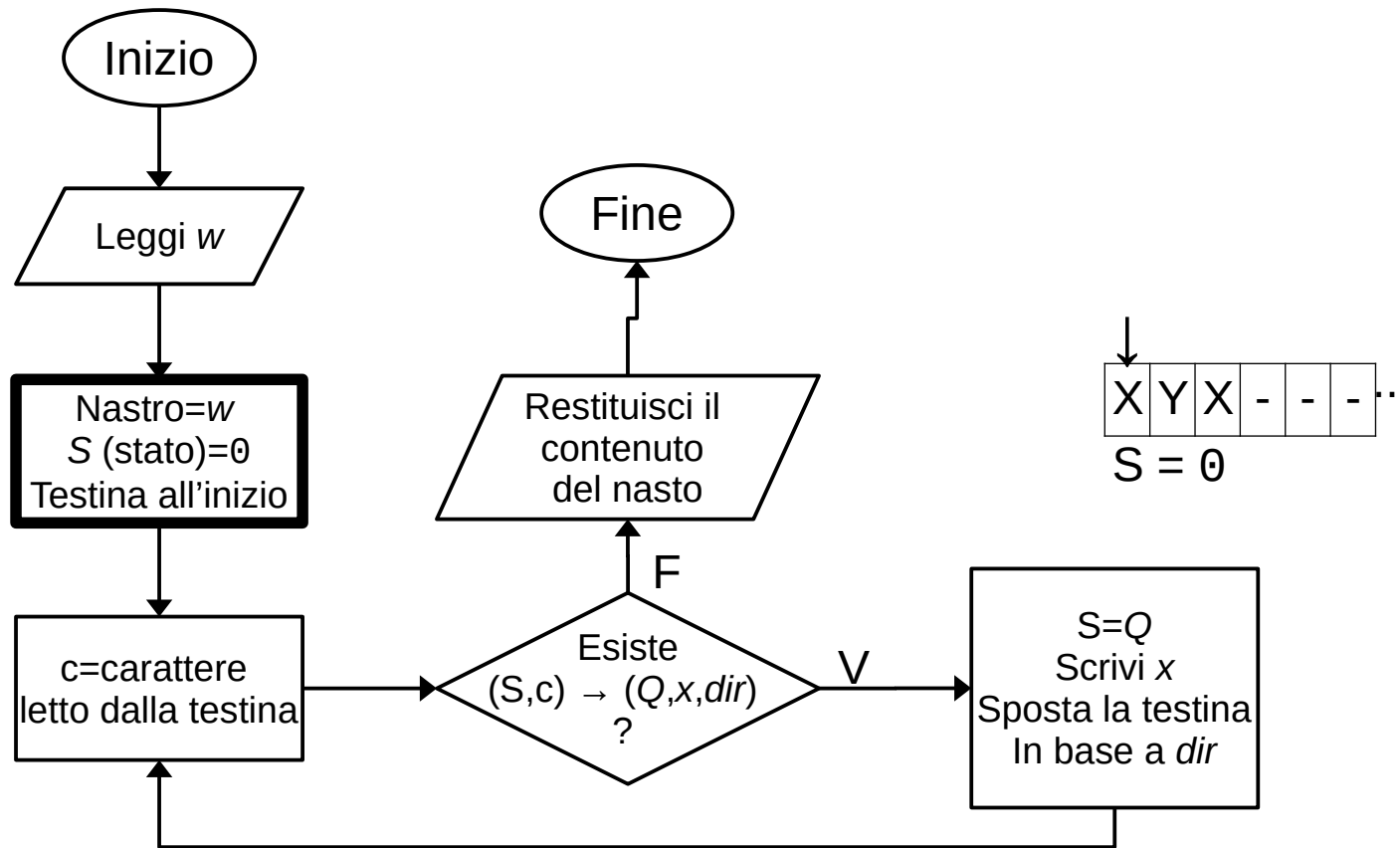
$(0, Y) \rightarrow (0, Y, >)$

Macchine di Turing – esempio 1 - esecuzione

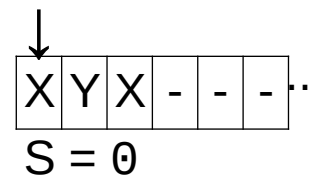


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

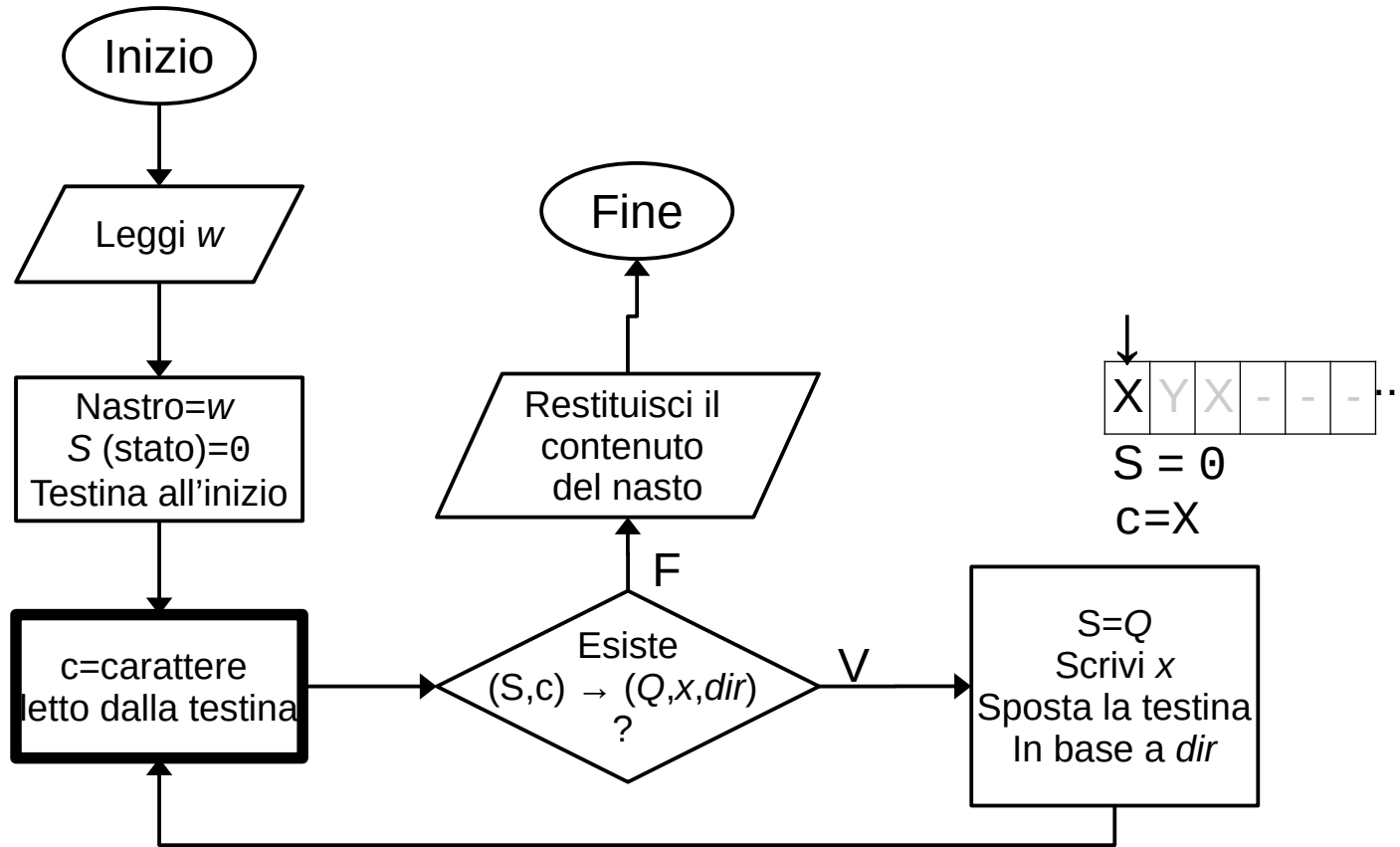
Macchine di Turing – es1 – configurazione iniziale



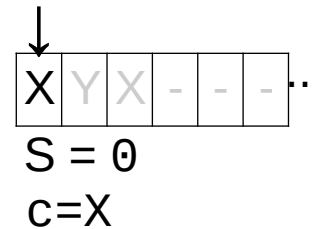
Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$



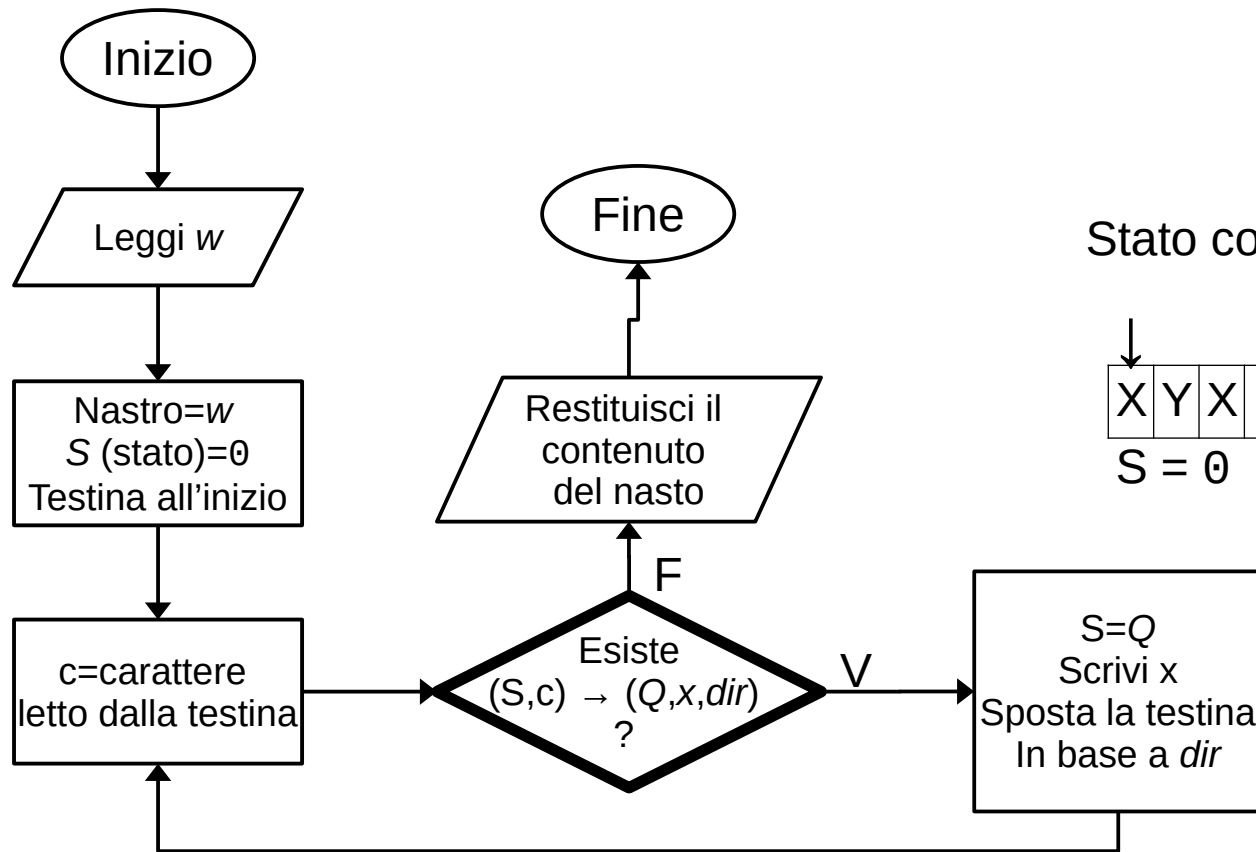
Macchine di Turing – es1 - esecuzione



Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

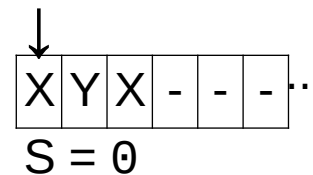


Macchine di Turing – es1 – esecuzione



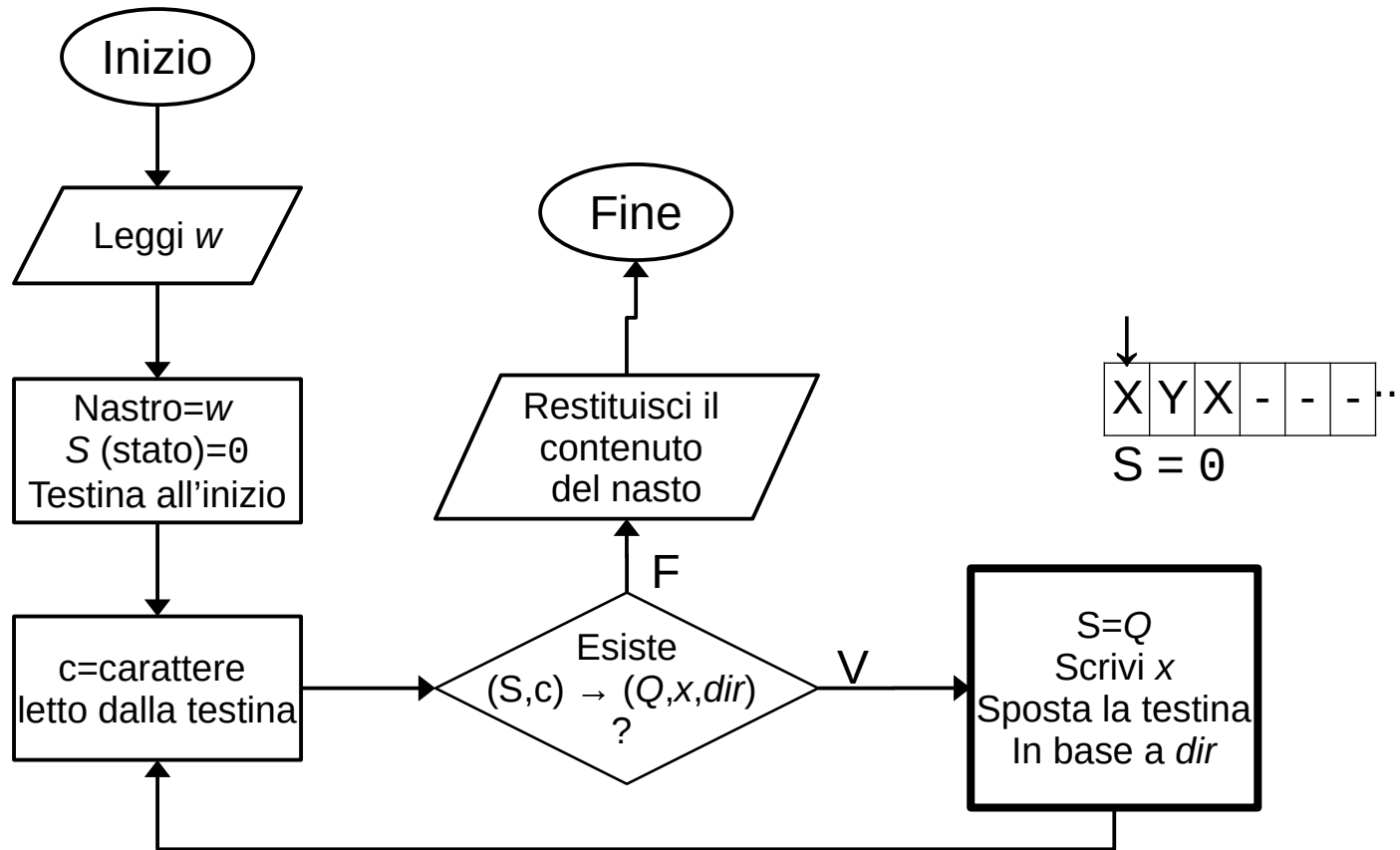
Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

Stato corrente

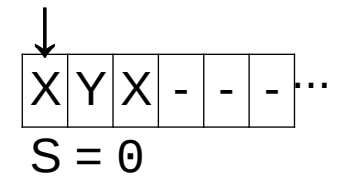


Simbolo letto

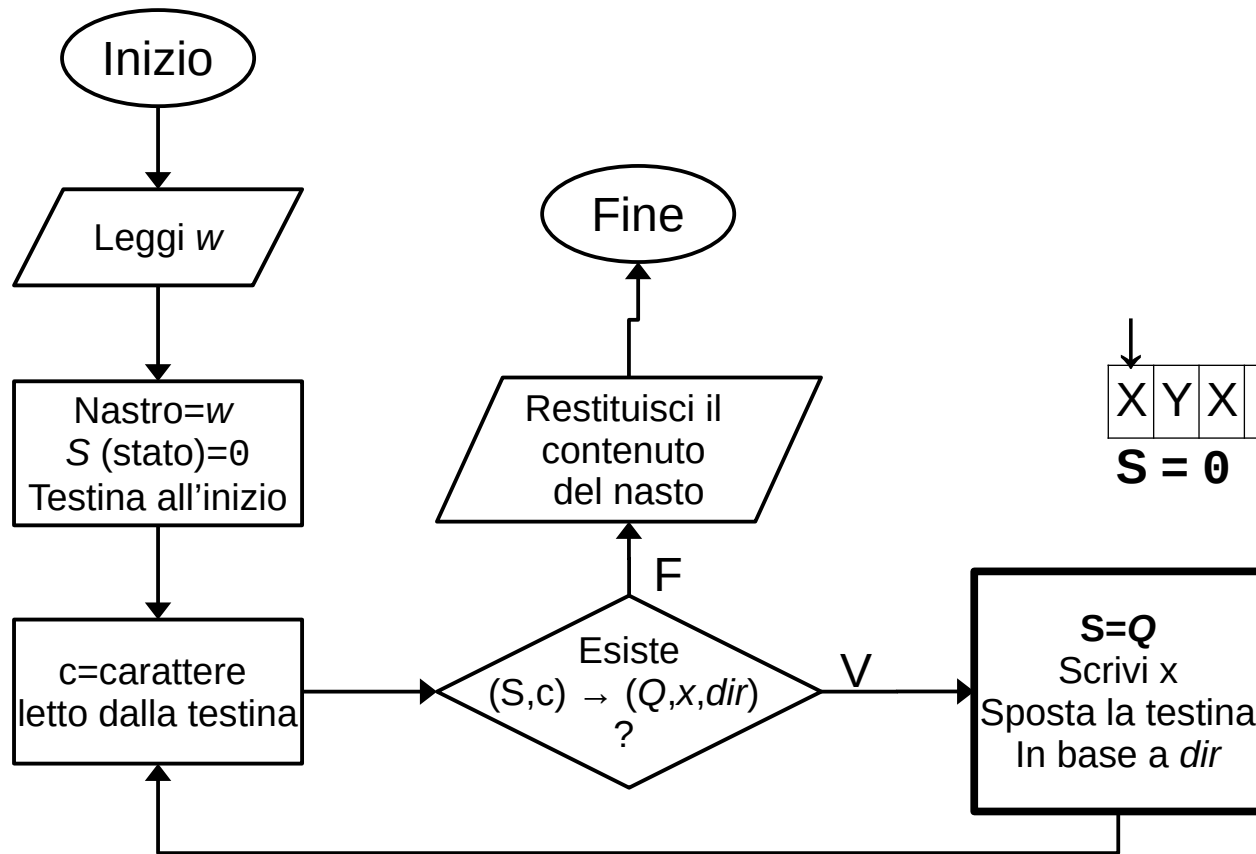
Macchine di Turing – es1 – esecuzione



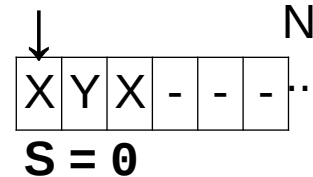
Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$



Macchine di Turing – es1 – esecuzione

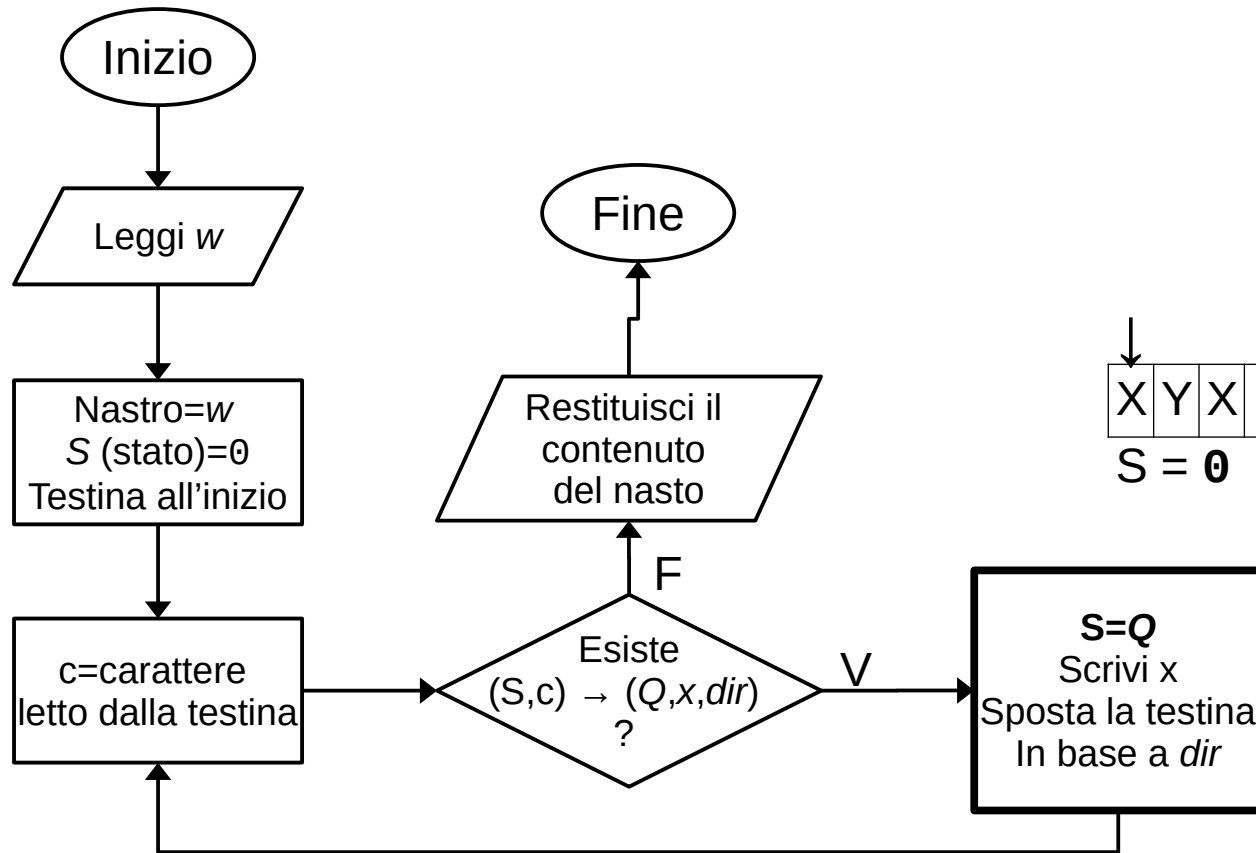


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

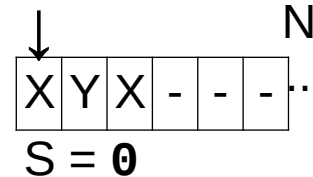


Nuovo stato

Macchine di Turing – es1 – esecuzione

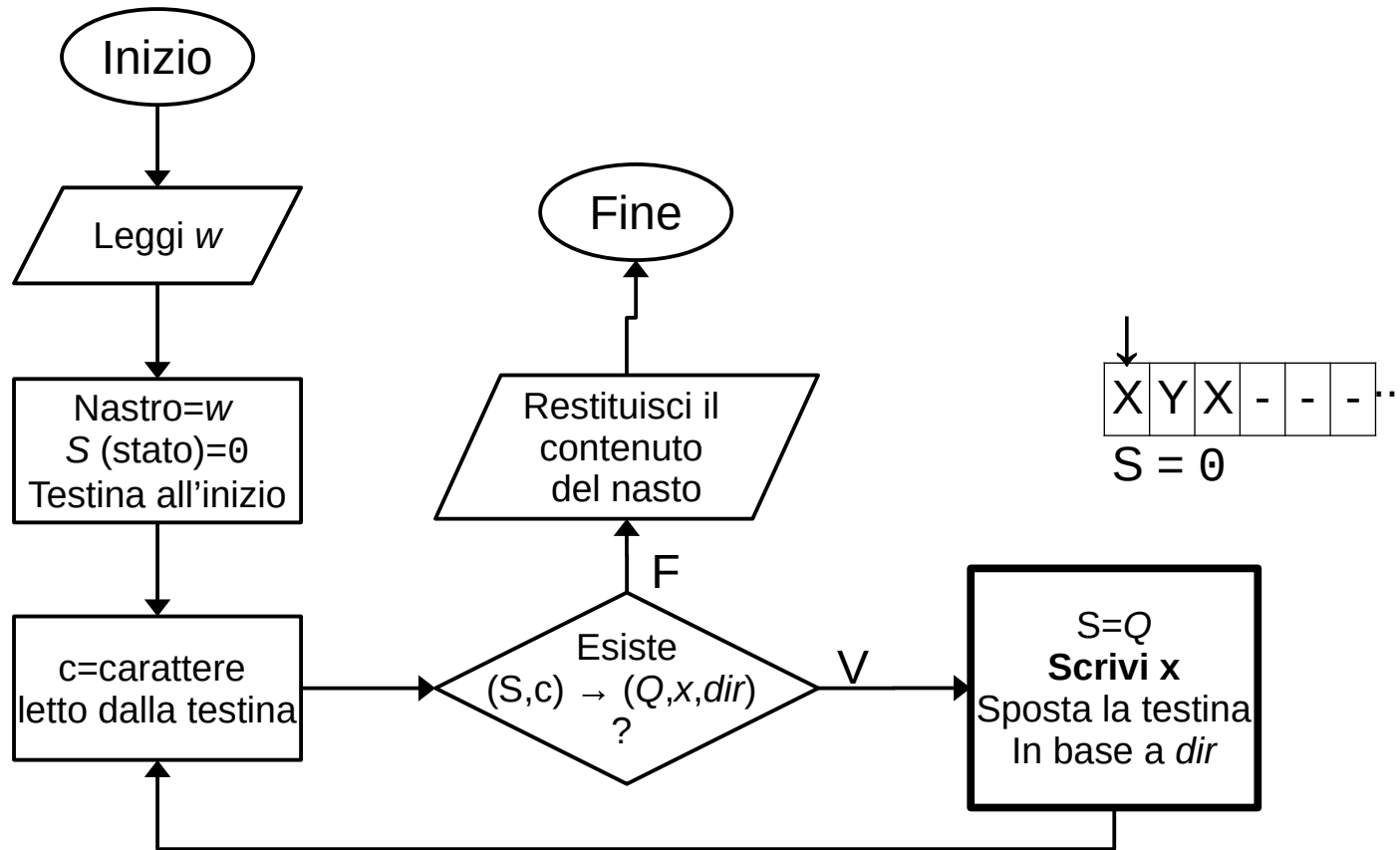


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

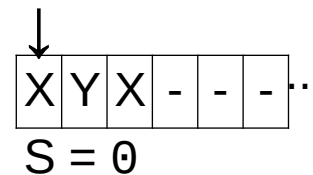


Nuovo stato

Macchine di Turing – es1 – esecuzione

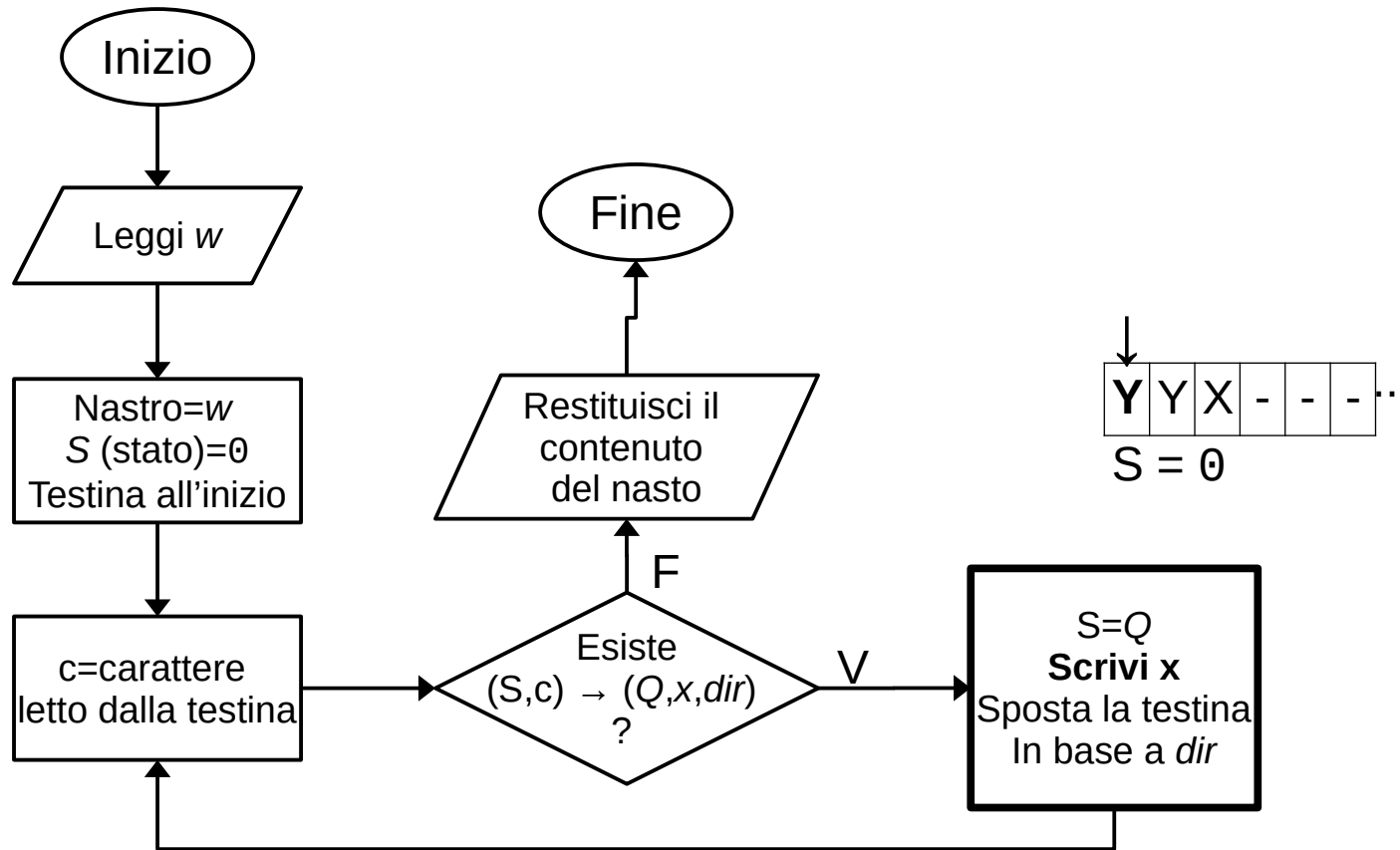


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

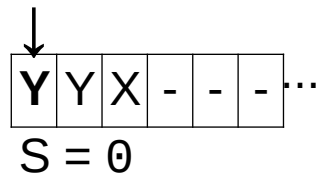


Simbolo da scrivere

Macchine di Turing – es1 – esecuzione

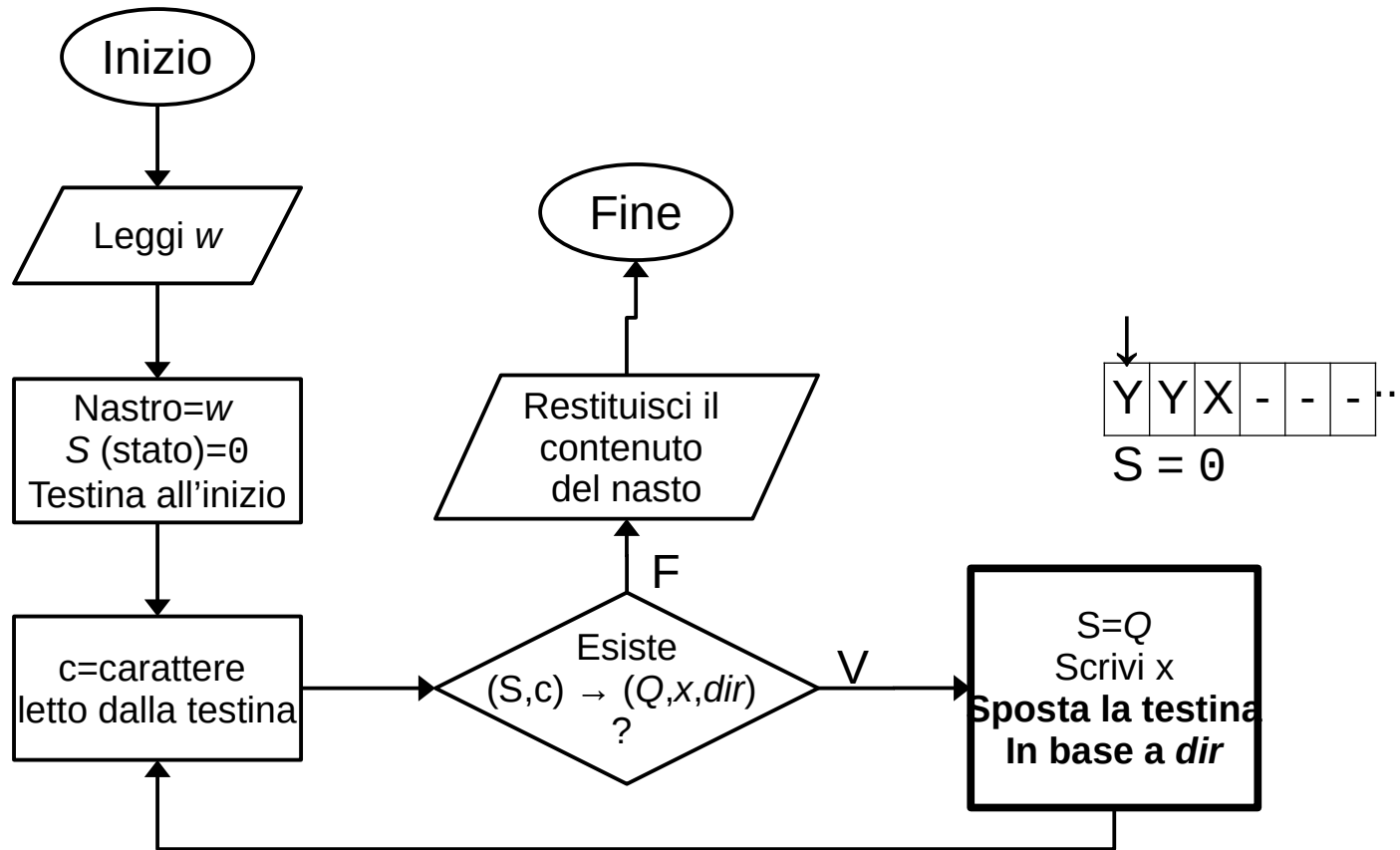


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

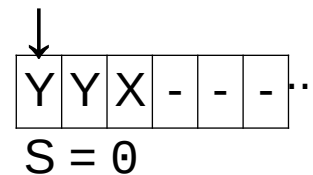


Simbolo da scrivere

Macchine di Turing – es1 – esecuzione

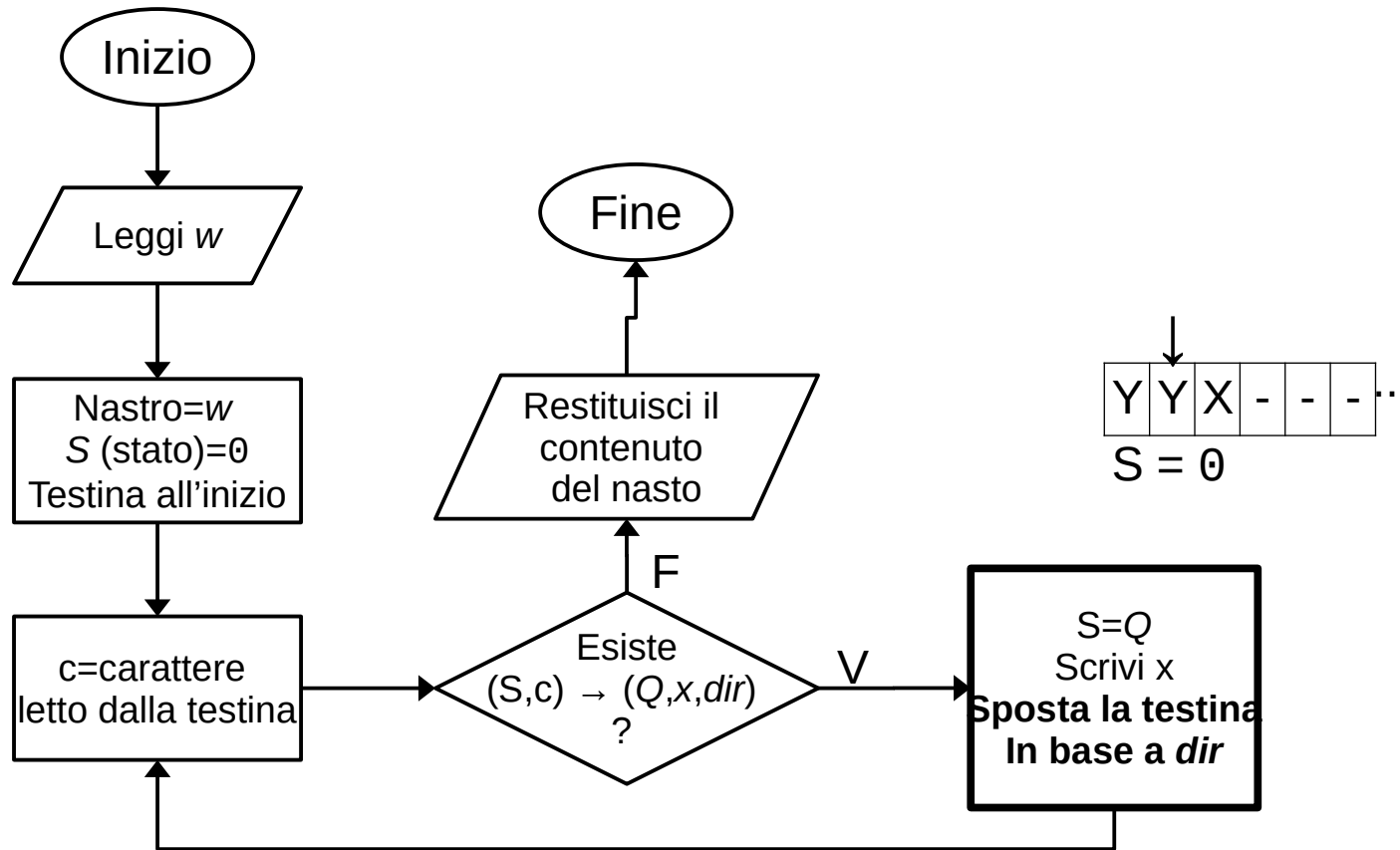


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

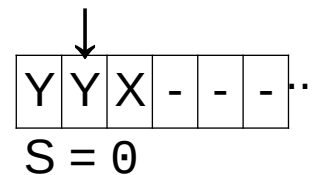


Spostamento

Macchine di Turing – es1 – esecuzione

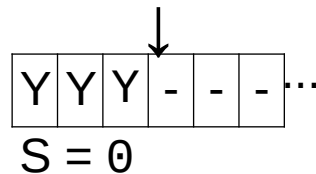
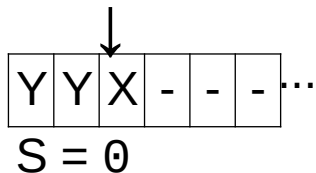
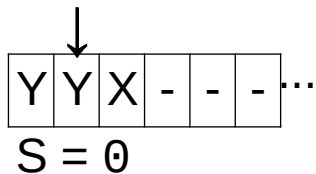


Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$



Spostamento

Macchine di Turing – es1 – esecuzione

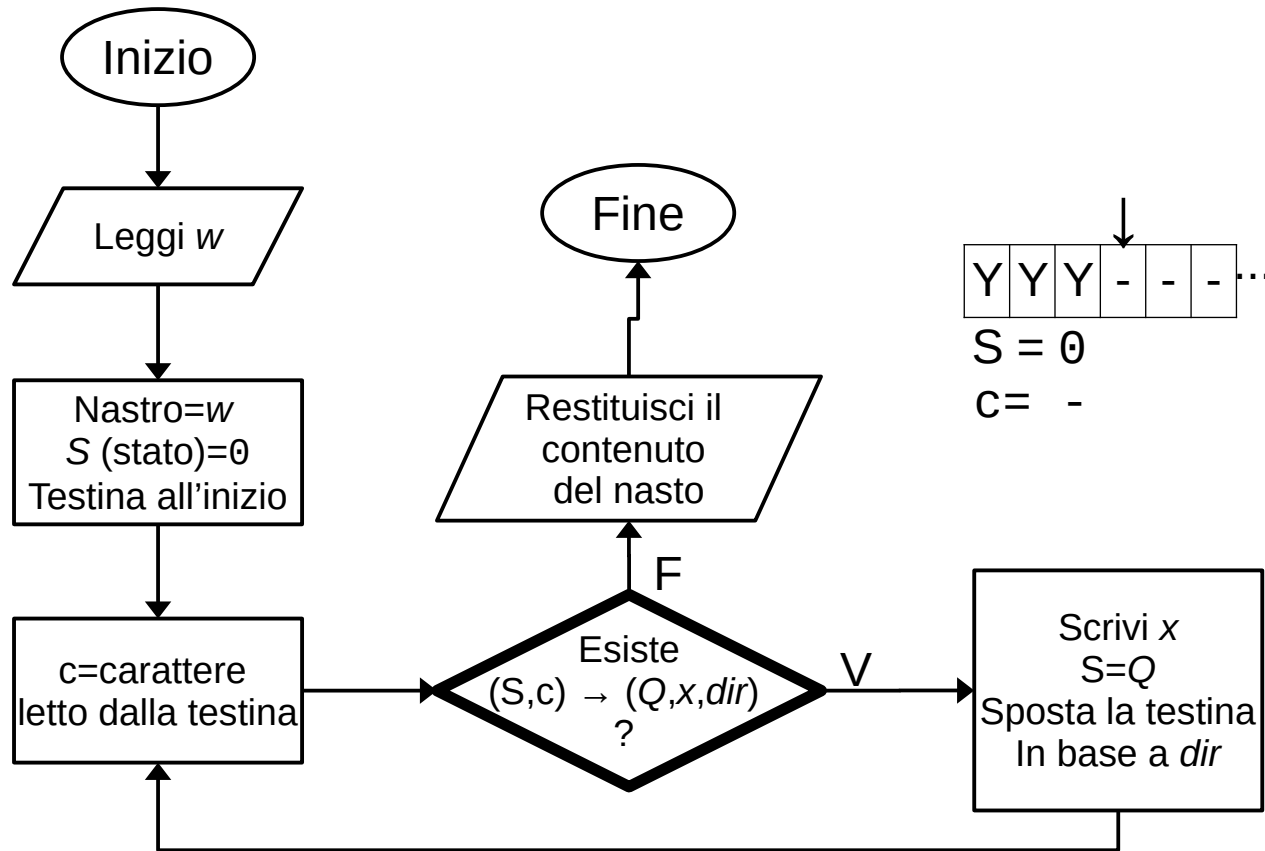


Regole di transizione

$(0, X) \rightarrow (0, Y, >)$

$(0, Y) \rightarrow (0, Y, >)$

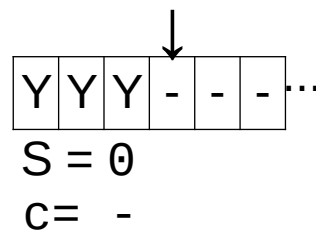
Macchine di Turing – es1 – esecuzione



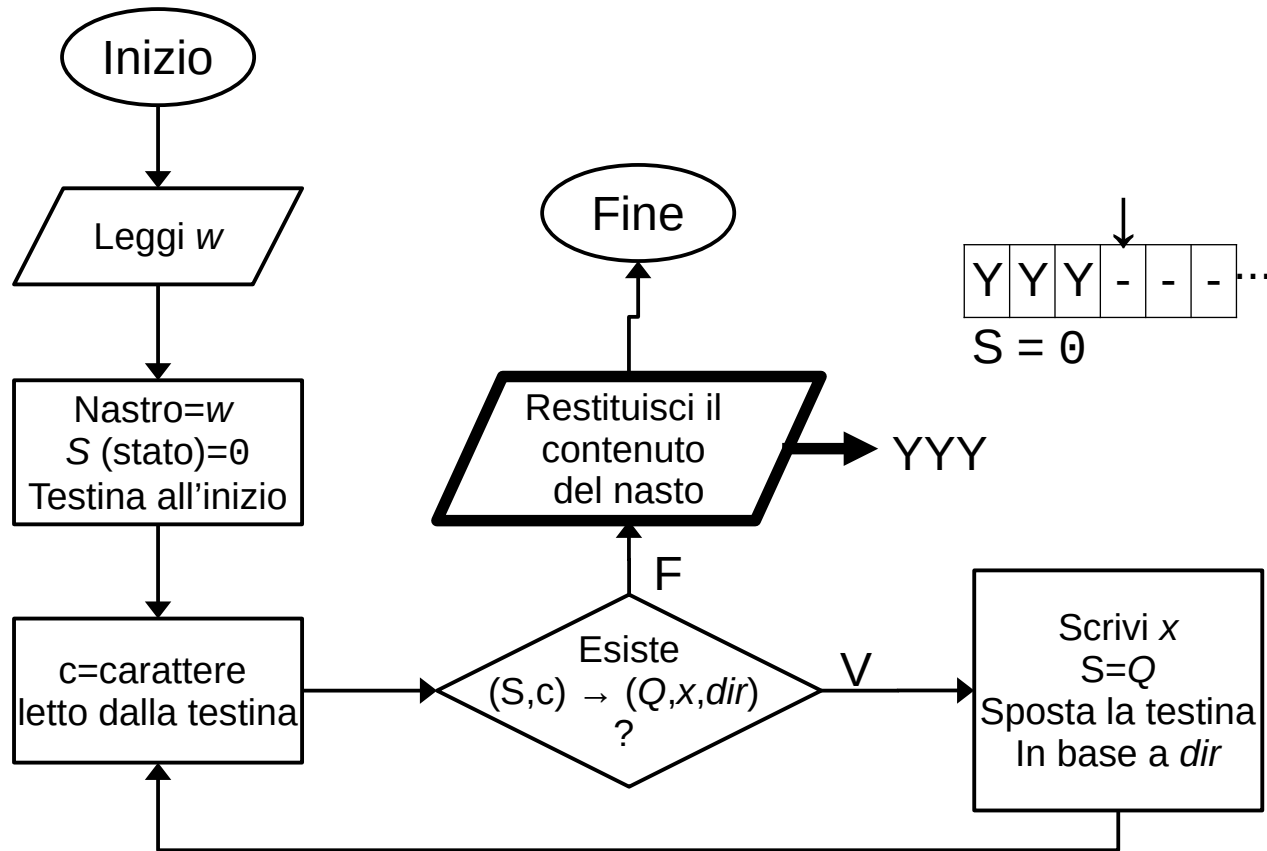
Regole di transizione

$(0, X) \rightarrow (0, Y, >)$

$(0, Y) \rightarrow (0, Y, >)$



Macchine di Turing – es1 – esecuzione



Regole di transizione
 $(0, X) \rightarrow (0, Y, >)$
 $(0, Y) \rightarrow (0, Y, >)$

Simulatore di Macchine di Turing

<https://www.turingsimulator.net/>

Alfabeto Γ : qualsiasi simbolo, anche il blank -

Stati Q : qualsiasi stringa di caratteri

Stato iniziale: sempre \emptyset

Regole di transizione: specificate come

(stato_corrente, carattere_letto, nuovo_stato, carattere_scritto, direzione)

Esempi di regole di transizione

$(\emptyset, X, \emptyset, Y, >)$

$(DISPARI, X, PARI, X, >)$

Macchine di Turing – es. 2

Problema: data una *stringa* di caratteri X e Y, se il primo carattere è X sostituire il secondo carattere con C. Altrimenti, se il primo carattere è Y, sostituire il secondo carattere con D.

Esempi

Dati di ingresso	Valore restituito
XYYXY	XCXYX
XXYXY	XCXYX
YYXY	YDXY
YXXY	YDXY
X	XC
Y	YD

Macchine di Turing – es. 2

Problema: data una *stringa* di caratteri X e Y, se il primo carattere è X sostituire il secondo carattere con C. Altrimenti, se il primo carattere è Y, sostituire il secondo carattere con D.

Esempi

Dati di ingresso	Valore restituito
XYYXY	XCXYX
XXYXY	XCXYX
YYXY	YDXY
YXXY	YDXY
X	XC
Y	YD

Regole di transizione

$(\emptyset, X) \rightarrow (\text{FIRSTX}, X, >)$

$(\emptyset, Y) \rightarrow (\text{FIRSTY}, Y, >)$

$(\text{FIRSTX}, X) \rightarrow (\text{END}, C, -)$

$(\text{FIRSTX}, Y) \rightarrow (\text{END}, C, -)$

$(\text{FIRSTX}, -) \rightarrow (\text{END}, C, -)$

$(\text{FIRSTY}, Y) \rightarrow (\text{END}, D, -)$

$(\text{FIRSTY}, Y) \rightarrow (\text{END}, D, -)$

$(\text{FIRSTY}, -) \rightarrow (\text{END}, D, -)$

Macchine di Turing – es. 3

Problema: data una *stringa* di caratteri X e Y, se il primo carattere è X sostituire tutte le Y con C. Altrimenti, se il primo carattere è Y, sostituire tutte le Y con D.

Esempi

Dati di ingresso	Valore restituito
XYYXY	XCCXC
YYXY	DDXD
Y	D
XXX	XXX
XYX	XCX

Macchine di Turing – es. 3

Problema: data una *stringa* di caratteri X e Y, se il primo carattere è X sostituire tutte le Y con C. Altrimenti, se il primo carattere è Y, sostituire tutte le Y con D.

Esempi

Dati di ingresso	Valore restituito
XYYXY	XCCXC
YYXY	DDXD
Y	D
XXX	XXX
XYX	XCX

Regole di transizione

$(\emptyset, X) \rightarrow (\text{FIRSTX}, X, >)$

$(\emptyset, Y) \rightarrow (\text{FIRSTY}, D, >)$

$(\text{FIRSTX}, X) \rightarrow (\text{FIRSTX}, X, >)$

$(\text{FIRSTX}, Y) \rightarrow (\text{FIRSTX}, C, >)$

$(\text{FIRSTY}, X) \rightarrow (\text{FIRSTY}, X, >)$

$(\text{FIRSTY}, Y) \rightarrow (\text{FIRSTY}, D, >)$

Macchine di Turing – es. 4

Problema: data una *stringa* di caratteri X e Y, sostituire C ad ogni X che è preceduto da Y.

Esempi

Dati di ingresso	Valore restituito
XYYXY	XYYCY
YXXY	YCX Y
XYXYYXXXXX	XYCYYCXXXX

Regole di transizione

$(\emptyset, X) \rightarrow (\text{PREDX}, X, >)$

$(\emptyset, Y) \rightarrow (\text{PREDY}, Y, >)$

$(\text{PREDX}, X) \rightarrow (\text{PREDX}, X, >)$

$(\text{PREDX}, Y) \rightarrow (\text{PREDY}, Y, >)$

$(\text{PREDY}, X) \rightarrow (\text{PREDX}, C, >)$

$(\text{PREDY}, Y) \rightarrow (\text{PREDY}, Y, >)$

Conseguenze

Esistono problemi **indecidibili**.

Conseguenze

Esistono problemi **indecidibili**.

Distinzione tra *hardware* e *software*.

Conseguenze

Esistono problemi **indecidibili**.

Distinzione tra *hardware* e *software*.

Architettura di von Neumann

