12/03/2024

LEZ 3

ALCUNI CONCETTI DEL PDF OF LEZ.2

∃ 2 TIPI DI MACCHINE DI TURING:

① TRASDUTTORE

- MACCHINA CON UN NUMERO ∞ DI RISULTATI

ESEMPIO: MACCHINA CHE FA SOMMA DI 2 NUM.
(∞ POSSIBILI RISULTATI).

- $F: \Sigma^* \to \Sigma^*$ → LA FUNZIONE CALCOLA ∞ RISULTATI
- RESULT → NASTRO OUTPUT → LA MACCHINA NEED IT
- SERVE 1 SOLO STATO : QUANDO HO FINITO IL CALCOLO
  q_F

② RICONOSCITORE

- MACCHINA CON N. COST. DI RISULTATI

ESEMPIO = MACCHINA CHE CONTROLLA SE IL NUM. DI 0 È PARI → (2 POSSIBILI SOLUZIONI)

- $f: \Sigma^* \to \{0,1\}$ CALCOLA N. COST. DI SOLUZIONI
(IN QUESTO CASO SE LA PAROLA HA PROPRIETÀ 0 OPP. 1)

- NO NASTRO OUTPUT → NON NE HA BISOGNO
  ↳ SERVE SOLO 1 STATO INTERNO
  $Q_F = \{q_A, q_R\}$ → STATI "RICONOSCITORI"

↳ q_A = SE PAROLA HA PROPRIETÀ
q_R = SE NON CE L'HA.

ESITO DELLA COMPUTAZIONE : $O_T(x)$

CAMBIA SE T È RICONOSCITORE O TRASDUTTORE

- $O_T(x)$ (TRASDUTTORE) = NASTRO OUTPUT
- $O_T(x)$ (RICONOSCITORE) = STATO INTERNO AFTER COMPUTAZIONE

N.B.
D'ORA IN POI X MACCHINA DI TURING SI INTENDE LA MACCHINA TYPE RICONOSCITORE

ARGO PDF 2 FINISH

TEOREMI SU M.d.t.

ARGO. PDF 3

1° TEOREMA

TEOREMI DELLE SIMULAZIONI

- $\forall$ T (m.d.t) CON k NASTRI COST. E TESTINE INDIPENDENTI,
  (TESTINE: si muovono dove gli pare)

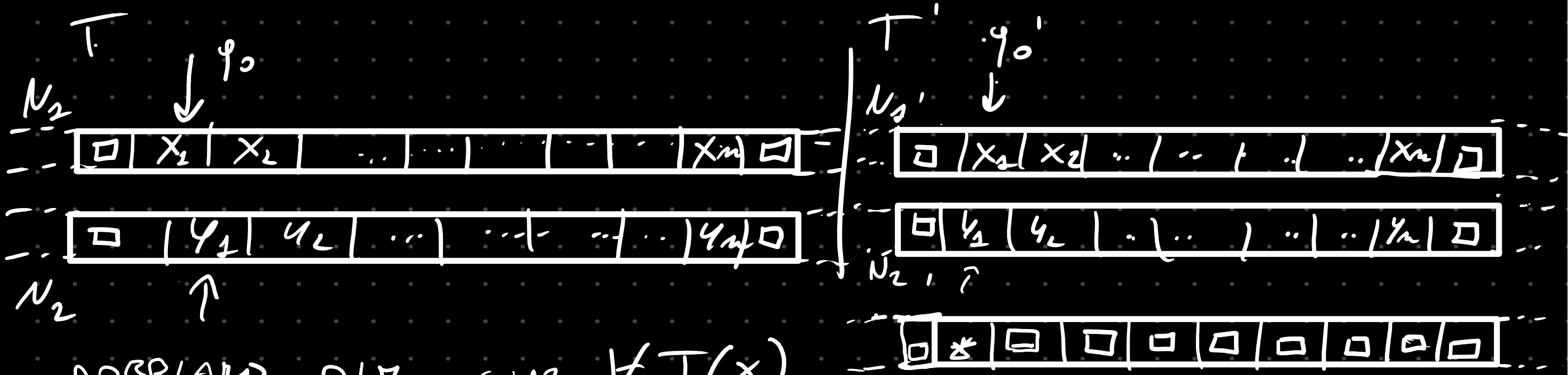
ALLORA

$\exists$ T' a $\underbrace{k+1}$ NASTRI E TESTINE SOLIDALI
(SOLIDALI: DEVONO MUOVERSI IN = DIREZIONE.)

DOVE $\underbrace{\forall x \in \Sigma^*}$ => $\underbrace{O_T(x) = O_{T'}(x)}$ → L'ESITO È UGUALE

DIM

CONSIDERIAMO LE 2 MACCHINE



DOBBIAMO DIM. CHE $\forall T(x)$ DI T PUÒ ESSERE 'SIMULATO' DA T'

dimostrazione per open box, cioè a 'scatola aperta'

CASO MOVIMENTO UNI-DIREZIONALE O FERMO

IN T,

$$<q_0, (x_1, y_1), (v, t), q, (d, d)>$$

IN T': LA COMPUTAZIONE RIMANE PRESSOCCHÈ INVARIATO CON AGGIUNTA UNA COMP. X SPOSTARE *.

$$① <q_0, (x_1, y_1, *), (v, t, \square), q^*, (d, d)>$$

$$② <q^*, (x, y, \square), (x, y, *), q_0, (F, F)>$$

N.B.
- X MOV = SINISTRA UGUALE, CAMBIA LA DIREZIONE
- X MOV = FERMO, QUINTUPLA IN T' = A T CON AGGIUNTO * (CHE NON VIENE SOVRASCRITTO)

q^* = STATO NUOVO X INDICARE DI RISCRIVERE *

MOV. BI-DIREZIONALE

IDEA: NON POTENDO SPOSTARSI IN DIREZIONI ≠, WE CAN IMAGINE I NASTRI CHE VENGONO "TRASCINATI" IN 2 DIR. ≠ COSÌ DA "SPOSTARE" I CONTENUTI DEI NASTRI SENZA SPOSTARE LE TESTINE.
IL CARATTERE '*' SERVE X RICORDARE LA POSIZIONE DOVE LE TESTINE DEVONO STARE.

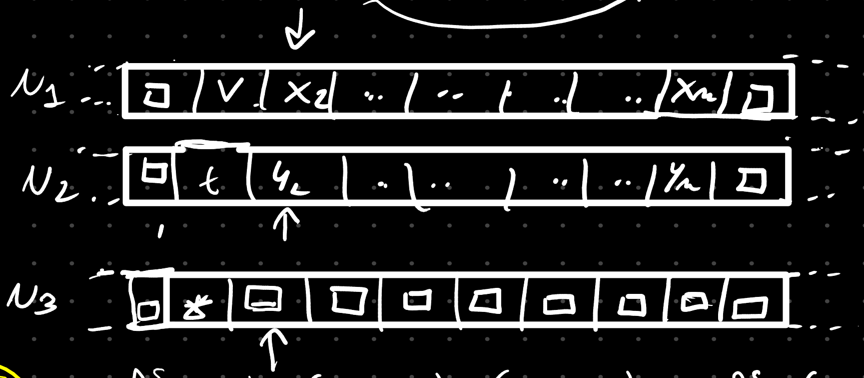
SIA QUINDI PER T: $<q_0, (x_1, y_1), (v, t), q_1, (d, s)>$

CASO DESTRA - SINISTRA -> (SINISTRA - DESTRA QUASI IDENTICO)

1 $\langle q_0, (x_1, y_1, *), (V, t, *), q^{DS}_{0D}(q_1), (d, d) \rangle \rightarrow q^{DS}_{0D}$ -> CASO ROUTINE

N. STATO DOVE CI TROVIAMO

DIR. SHIFT



2 $\langle q^{DS}_{0,D}(q_1), (x,y,z), (x,y,z), q^{DS}_{0,D}(q_1), (d, d) \rangle$

$$\forall x \in \Sigma, y \in \Sigma \cup \{\square\} \ \forall z \in \{*, \square\}$$

sposta tutte le testine nella pos più a destra del primo nastro

3 $\langle q^{DS}_{0D}(q_1), (\square, y, z), (\square, y, z), q^{DS}_{0,S}(q_1, \square), (s, s) \rangle$

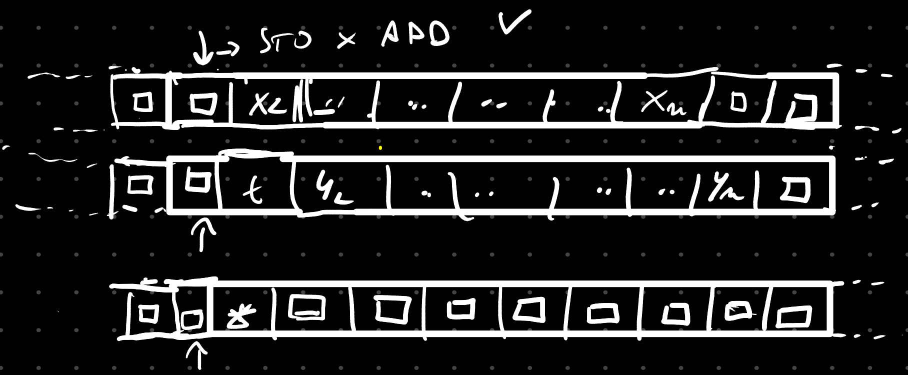
$$\forall y \in \Sigma \cup \{\square\} \ \forall z \in \{*, \square\}.$$

prepara il left-shift

4 $\langle q^{DS}_{0,S}(q_1, a), (x, y, z), (a, y, z), q^{DS}_{0,S}(q_1, x), (s, s, s) \rangle$

$$\forall a \in \Sigma \cup \{\square\}, \forall z \in \{*, \square\}, \forall x, y \in \Sigma$$

SHIFT-SINISTRO -> (scrivo a, cancello x e me lo ricordo (nuovo stato, per ogni possibile carattere letto), e mi sposto a sinistra. repeat



(5) $\langle q^{DS}_{D,S}(q_1, a), (\square, \square, z), (a, \square, z), q^{DS}_{1,D}(q_1, \square), (d, d, d) \rangle$

$\forall a \in \Sigma \cup \{\square\}\ \forall z \in \{*, \square\}$

arrivato alla fine del primo nastro (sta per aggiungere a), prepara il right-shift per il secondo nastro.

(6) $\langle q^{DS}_{1,D}(q_1, b), (x, y, z), (x, b, z), q^{DS}_{1,D}(q_1, y), (d, d, d) \rangle$

$\forall b \in \Sigma \cup \{\square\},\ \forall z \in \{*, \square\},\ \forall y \in \Sigma,\ \forall x \in \Sigma$

right shift

(7) $\langle q^{DS}_{1,D}(q_1, b), (x, \square, z), (x, b, z), q^{DS}_{1,D}(q_1), (f, f, f) \rangle$

$\forall b \in \Sigma \cup \{\square\}\ \forall x \in \Sigma \cup \{\square\}\ \forall z \in \{*, \square\}$

raggiunto l'ultimo carattere del nastro.
begin spostamento nella cella dove si trova *.

(8) $\langle q^{DS}_{1,S}(q_1), (x, y, \square), (x, y, \square), q^{DS}_{1,S}(q_1), (s, s, s) \rangle$

(9) $\langle q^{DS}_{1,S}(q_1), (x, y, *), (x, y, *), q_1, (f, f, f) \rangle$

sis sposta fino a trovare *. quando ha finito va nello stesso q1, la stessa della quintupla eseguita da T1.

2° TEOREMA

$\forall$ T a k nastri
$\exists$ T' a 1 nastro: $\forall x \in \Sigma^* \left[O_T(x) = O_{T'}(x) \right]$

DIM → = a quello sopra (simulazione)

- Sia T_3 a 3 nastri a testine solidali → indipend. va bene lo stesso ⇓ DIM SOPRA
  ↳ in (generale k)

T

□	x_1	x_2	..	..	..	..	..	..	..	..	x_n	□

□	y_1	y_2	..	....	.	..	..	.	y_m	□

□	z_1	z_2	..	....	.	..	.	...	z_n	□	□

Consideriamo T' con:
- $= \Sigma$ di T → num. di nastri k
- $Q = Q \times \Sigma^k$

E con N_1 scritto così:

□	x_1	y_1	z_1	x_2	y_2	z_2	x_3	y_3	z_3	x_4	y_4	z_4	..	..	..	.	.	.	..	x_n	y_n	z_n	□

una concatenazione di tutti i simboli dei k nastri, a partire dalla cella iniziale 1

COMPUTAZIONE

Sia la computazione in T_3

$$\langle q_0, (x_1, y_1, z_1), (a, b, c), q_1, (d, d, d) \rangle$$

In T_3 è solo questa, ma x T' need + quintuple:

① T' deve leggere k simboli (in this case 3) e devono essere in ordine (x_1, y_1, z_1). Quindi T' controlla se x_1 è seguito da y_1 etc.

QUINDI

$$\langle q_0, X_1, X_1, q(q_0,X_1), d\rangle$$
$$\to \langle q(q_0,X_1), Y_1, Y_1, q(q_0,X_1,Y_1), d\rangle$$
$$\to \langle q(q_0,X_1,Y_1), Z_1, Z_1, q(q_0,X_1,Y_1,Z_1), S\rangle$$

LETTO ULTIMO SIMB. PREP. FOR

con k nastri, controlla che i primi k simboli dei nastri siamo quelli.

② LETTO LAST SYMBOL, VA AL SIMBOLO DI 1° NASTRO

$$\langle q(q_0,x_1,y_1,z_1), Y_1, Y_1, q(q_0,x_1,y_1,z_1,2), S\rangle$$

↳ IN K NASTRI, GO LEFT FINO A $q(q_0, s_{1_1}, s_{1_2}, \dots, s_{1_k}, 1)$

③ ARRIVATO AL SIMBOLO DI N_1, INIZIA LA SCRITTURA.

$$\langle q_0(q_0,x_1,y_1,z_1,1), x_1, a, q_w(q_0,x_1,y_1,z_1,2), d\rangle$$
$$\Downarrow$$

IN GENERALE

$$\langle q_w(q_0,s_{1_1},s_{1_2},\dots,s_{1_k},i), s_{1_i}, s_{2_i}, q_w(q_0,s_{1_1},\dots,s_{1_k},i+1), d\rangle \quad (i=1,\dots,k-1)$$

ARRIVATA ALLA FINE

$$\langle q_w(q_0,x_1,y_1,z_1,3), Z_1, c, q', m'\rangle$$

④ q' e m' DIPENDE DA m, DOVE:

- SE $m = d$, → LE TESTINE VANNO SUL 2° SIMBOLO IN T' (IN QUESTO CASO):

$$\langle q_w(q_0,x_1,y_1,z_1,3), Z_1, C, Y_1, d\rangle$$ → VA AL 2° SIMBOLO DI N_1.

- SE $m = F$, NUOVE QUINT. DI T' DOVE SPOSTO LA TESTINA A SINISTRA FINO PER K CELLE., CON DEI NUOVI STATI DI TRANSIZIONE, DOVE ALLA FINE METTENDO q_1

- SE $m = S$,
  NUOVE QU. DI T', DOVE CON STATI DI TRANSIZIONE
  MUOVE A SINISTRA PER $2k$ CELLE
  ↓
  STATO FINALE $= q_1$.

3° TEOREMA

$$\forall T \text{ con } |\Sigma| \geq 3, \exists T' \text{ con } \Sigma = \{0,1\} \text{ t.c.}$$
$$\forall x \in \Sigma, \text{ e } x' \in \Sigma' \Rightarrow O_T(x) = O_{T'}(x')$$

IN SOSTANZA:

- OGNI SIMBOLO DI Σ CODIFICATO IN BINARIO
- NUOVE QUINTUPLE → LEGGONO I SIMBOLI COD. IN BINARIO
  ↳ SIMILE A 2° TEOREMA