AUTOMI, CALCOLABILITÀ, COMPLESSITÀ

DANIELE VENTURI

Esame: Scritto. Tre parti.

Cosa faremo in questo corso. La domanda:
Quali sono le limitazioni INTRINSECHE
della computazione. Tre parti:

- Teoria degli AUTOMI. Il primo passo
per capire modelli semplici di computazione.
Applicazioni: Parser, compilatori, elaborazione

Testo per estrarre dati.

- Computabilità. Consideremo modello di calcolo molto più potente: la MACCHINA DI TURING. Esistono problemi che nessun computer può risolvere.
Esempio: Stabilire se programma termina. Indipendentemente dalle risorse!
Anche Teorema di GOEDEL.

- Complessità. Quanto finiremo le risorse: spazio e tempo. Sembrano esistere problemi DIFFICILI da risolvere in modo efficiente.

Esempio: FACTORING.

$n = p \cdot q$, p, q PRIMI

di n bit.

$n = 1024$

Calcolare n è facile (Tempo polinomiale nella dim. input). La classe P.

Fattorizzare n sembra difficile, con

i migliori algoritmi che hanno tempo (sub) exponential.

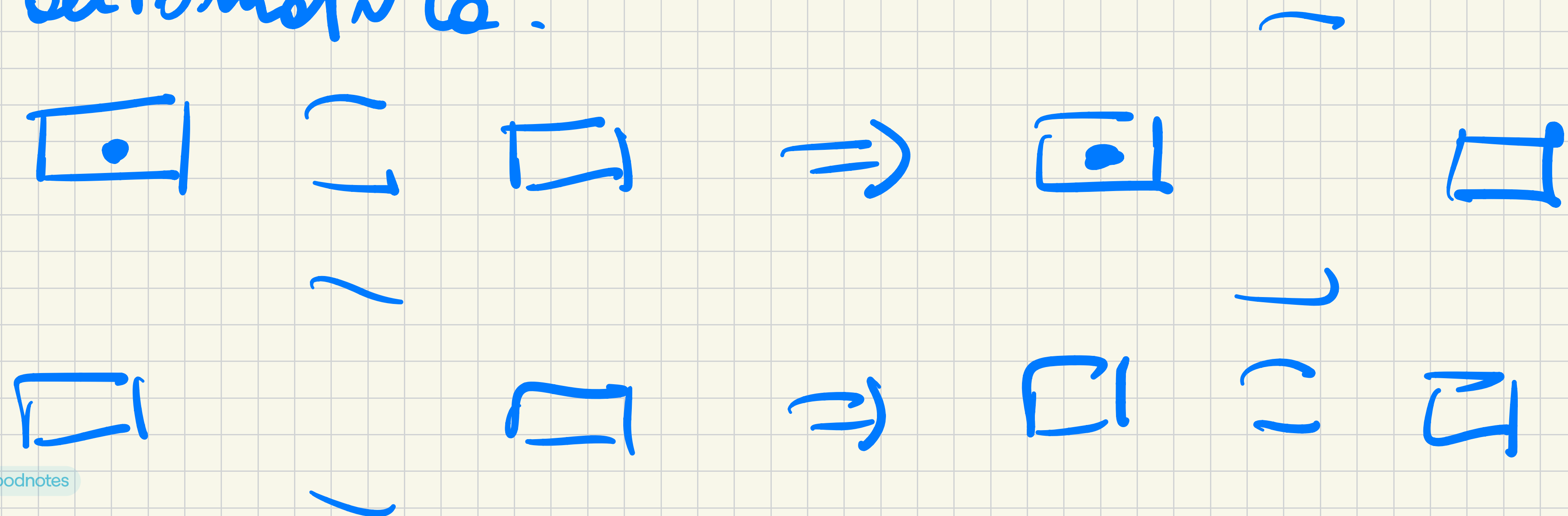
Osservazione: Dati p, q candidati (soluzione) posso controllare efficientemente se è corretta. Classe NP.

P = NP ???

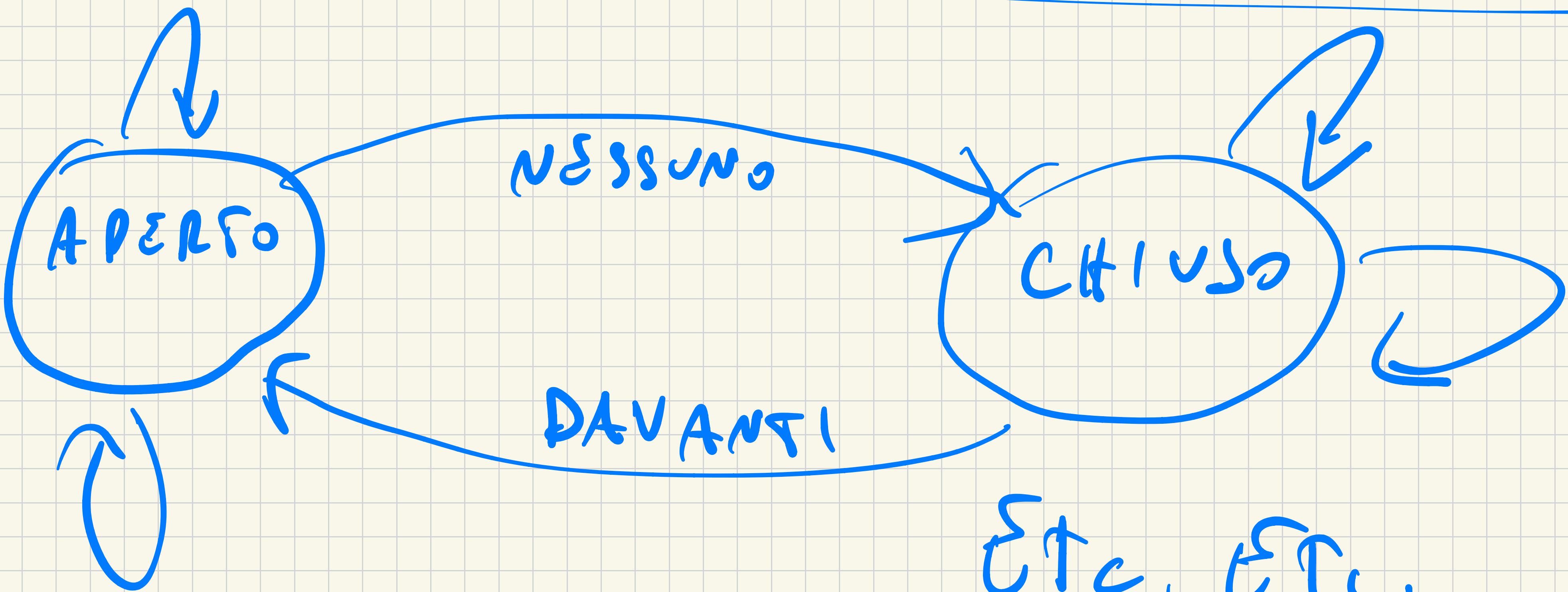
LINGUAGGI REGOLARI

Il primo modello: AUTOMA A STATI FINITI (DFA). Semplice: processa input bit a bit in modo sequenziale.

Esempio: Dispositivo apertura porta automatica.

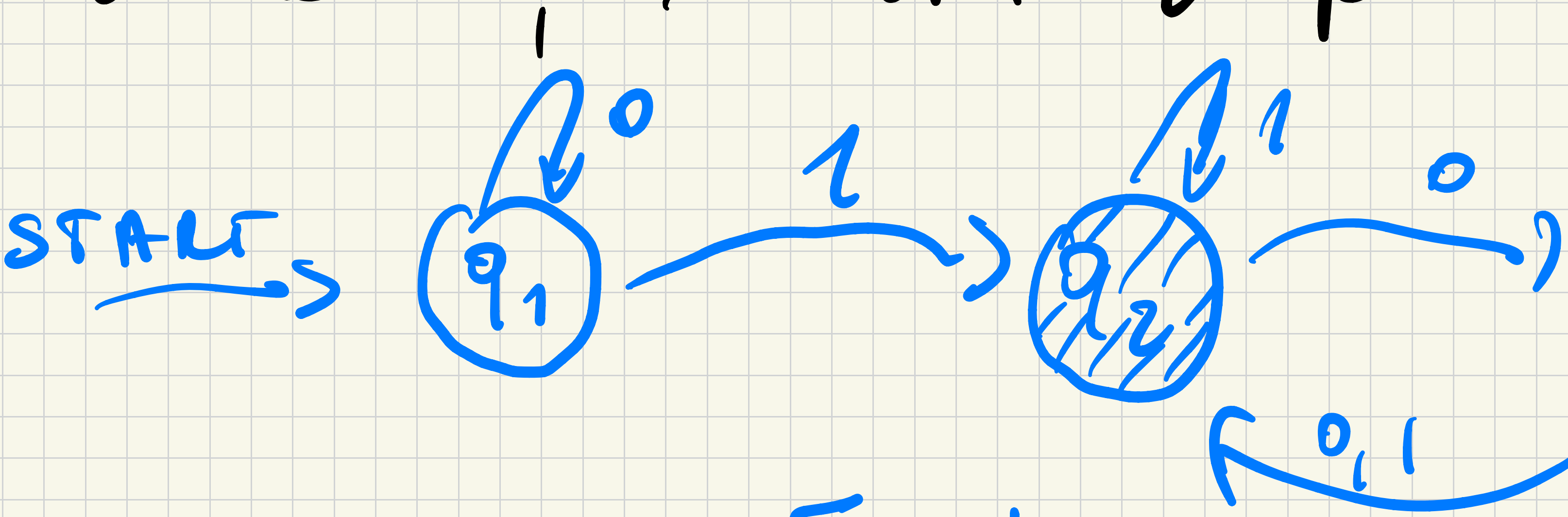


	NESSUNO	DAVANTI	DIETRO	ENTRAMBI
CHIUSO	CHIUSO	APERTO	...	...
APERTO	...	...	...	...



ETC, ETC.

Aggiornando un DFA è fatto così:



$\Sigma = \{0, 1\}$

$Q = \{q_1, q_2, q_3\}$

$w = 11101$ $\quad F = \{q_2\}$

$q_0 = q_1$

- q_1, q_2, q_3 sono STATI.
- q_2 è STATO ACCETTAZIONE.
- Archi sono TRANSIZIONI.

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

DEF (DFA). Un DFA è una tupla

$$(Q, \Sigma, \delta, q_0, F)$$

- Q insieme finito degli stati.
- Σ insieme finito dei simboli in input.
- $\delta : Q \times \Sigma \to Q$
- q_0 stato iniziale.
- $F \subseteq Q$ stati finali (ACCETTAZIONE)

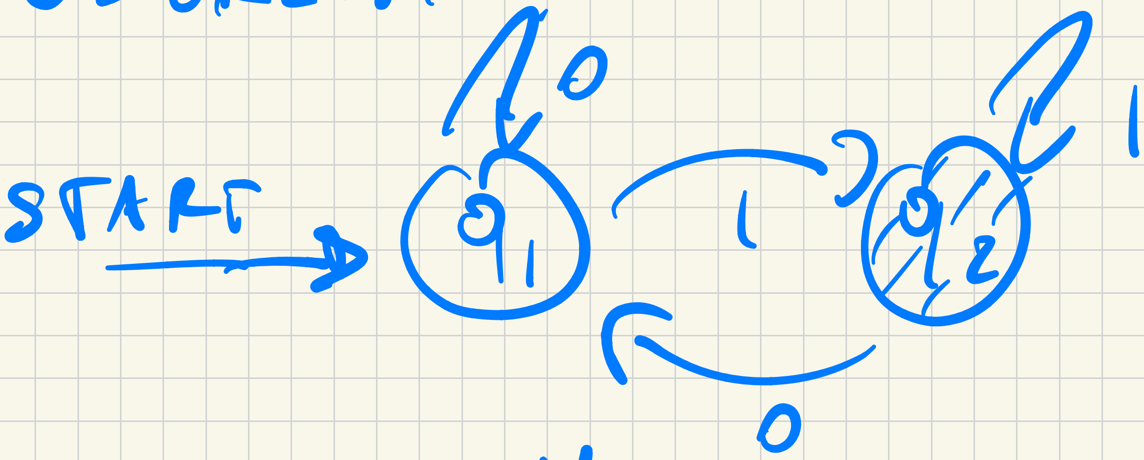
Se M è un DFA, l'insieme di stringhe riconosciute da M si denota

$$L(M)$$

ovvero il LINGUAGGIO riconosciuto da M.
(Può essere che $L = \emptyset$.)

Es: DFA precedente ha linguaggio delle stringhe w : w contiene almeno un "1" e # pari di "0" segue l'ultimo "1".

Esercizio:



M

$L(M) = ?$

Per definire precisamente il linguaggio introduciamo la FUNZIONE DI TRANSIZIONE ESTESA:

$$\delta^*: Q \times \Sigma^* \to Q$$

$$\begin{cases} \delta^*(q, \varepsilon) = \delta(q, \varepsilon) \\ \delta^*(q, ax) = \delta^*(\delta(q, a), x) \end{cases}$$

$x \in \Sigma^*,\ e \in \Sigma$

Altro concetto è la CONFIGURAZIONE

è coppia in $Q \times \Sigma^*$: (1) lo stato

e (2) cosa resta da leggere.

Dato $x \in \Sigma^*$, la configurazione iniziale

è (q_0, x).

Passo di computazione: parte da una configurazione ed un'altra rispettando la δ.

Relazione binaria:

$$(p, ax) \vdash_M (q, x) \quad \text{sse}$$

$$\delta(p, a) = q \quad \text{dove} \quad \begin{array}{l} p, q \in Q \\ a \in \Sigma \\ x \in \Sigma^* \end{array}$$

La posso estendere ($\vdash_M^*$) considerando la CHIUSURA riflessiva e transitiva:

(i) $(q, x) \vdash_M^* (q, x)$

(ii) $(q, aby) \vdash_M (p, by)$ e

$(p, by) \vdash_M (r, y)$

$\Rightarrow (q, aby) \vdash_M^* (r, y)$

$q, p, r \in Q$; $a, b \in \Sigma$; $y \in \Sigma^*$

DEF (LINGUAGGIO ACCETTATO). Diremo che $x \in \Sigma^*$ è accettato da

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

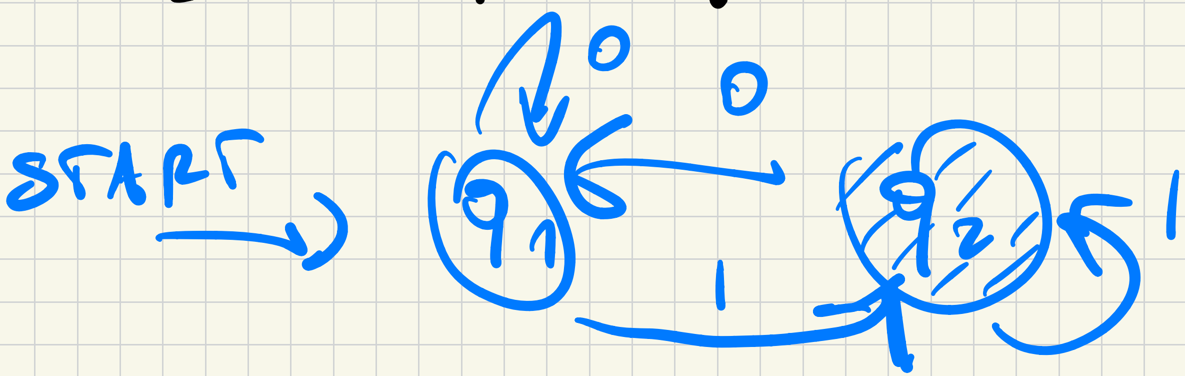
$$\delta^*(q_0, x) \in F \quad \text{oppure}$$

$$(q_0, x) \vdash_M^* (q, \varepsilon) \qquad q \in F.$$

STRINGA VUOTA

In altre parole:

$$L(M) = \{ x \in \Sigma^* : \delta^*(q_0, x) \in F \}$$



$$(q_1, 011) \vdash_M (q_1, 11) \vdash_M (q_2, 1)$$

$\varepsilon \equiv$ STRINGA VUOTA $\vdash_M (q_2, \varepsilon)$

$$\Rightarrow \quad \delta^*(q_1, 011) = q_2 \varepsilon)$$

$$(q_1, 011) \vdash_M^* (q_2, \varepsilon) \qquad q_2 \in F$$

$$w = 011 \in L(M)$$