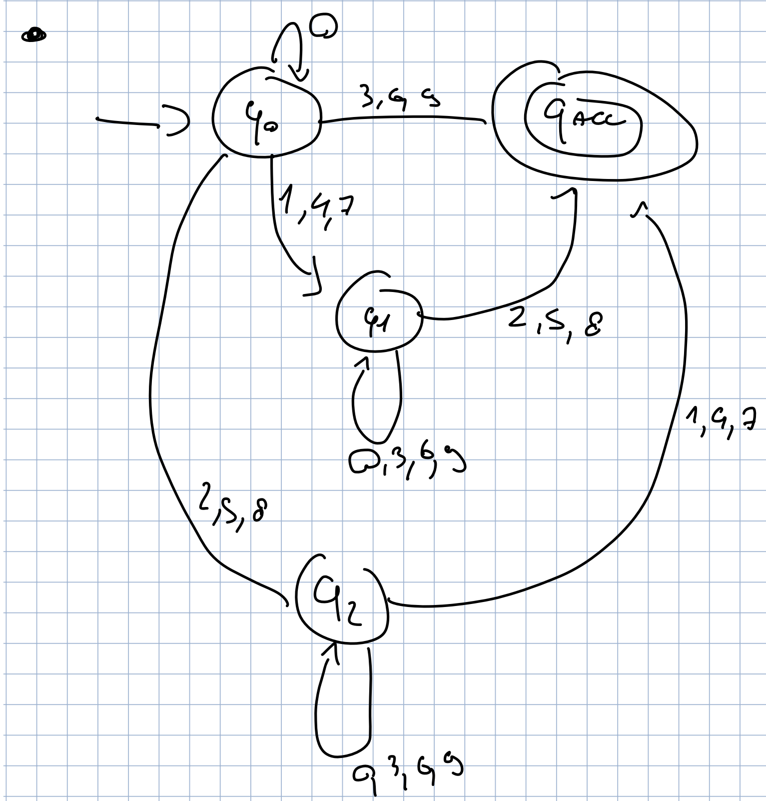
1 Automi — 10 Points

- Progettare un DFA che riconosca il linguaggio di stringhe su alfabeto $\Sigma = \{0, 1, \ldots, 9\}$ tali che la somma delle cifre sia un multiplo di 3. Provare la correttezza del DFA proposto.
- Enunciare e dimostrare il *pumping lemma* per linguaggi regolari. Fornire un esempio di applicazione.



é chiaro che il DFA funzion poiché accetta solo se la somma é multiplo di 3:

- 1° 3,6,9
- 2° 1,4,7 - 2,5,8 (sommano qualsiasi cosa) $x \% 3 = 0$
- 3° 2,5,8 - 1,4,7 uguale a sopra

- Il pumping lemma afferma che un $L \in REG$ HA un $p(|Q|)$ detto pumping $\forall w \in L$ t.c.
  - $|y| \neq 0$ con $w = xyz$
  - $|xy| \leq p$
  - $x y^i z \in L$

Dim: Sia $p = |Q|$ di un DFA che accetta $L \in REG$

sia $w = w_1 \dots w_n \in R = \{r_1 \dots r_n\}$ t.c.

$$\forall i \ \delta(r_i, w_i) = r_{i+1}$$

è chiaro che $n+1 > p \Rightarrow$ c'è uno stato ripetuto in R (successione stati) chiamiamo r_j la prima apparizione e r_l l'ultima

Dividiamo $w = xyz$ con

$x = w_1 \dots w_{j-1}$

$y = w_j \dots w_{l-1}$

$z = w_l \dots w_n$

è chiaro che $j \neq l \Rightarrow |y| \neq 0$, $|xy| \leq p$ e $x y^i z \in L$ poiché y ritorna in se stesso $\square$

Esempio: $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

sia $p = n \Rightarrow w = xyz$ prendo una scomposizione

$x = 0^p$

$y = 0^p 1^{n-q}$ è chiaro che $0^p 0^{(n-q)i} 1^q \notin L$

$z = 1^q$ $\square$

2 Calcolabilità 10 Points

- Si consideri il linguaggio W_{TM} delle stringhe della forma $\langle M, w, a\rangle$, dove M è una TM, w è una stringa appartenente all'alfabeto di input di M, ed a è un simbolo appartenente all'alfabeto di nastro di M, tale che M scrive a sul nastro quando è eseguita su input w. Mostrare che W_{TM} è indecidibile.
- Dimostrare che un linguaggio è Turing-riconoscibile se e solo se esiste una TM multi-nastro che lo riconosce.

- voglio ridurre W_{TM} a A_{TM}

$$A_{TM} \le_m W_{TM}$$

$\Rightarrow$ $f: \Sigma^* \to \Sigma^*$ t.c.

$$\forall \langle M,w\rangle \quad \langle M,w\rangle \in A_{TM} \iff f(\langle M,w\rangle) \in W_{TM}$$

Uso F come TM che calcola f

- input = $\langle M,w\rangle$
- simula $M(w)$
- se accetta ritorna $\langle M,w,a\rangle$ con a ultimo carattere scritto sul nastro
- se rifiuta simula D una TM che rifiuta sempre
  - ritorna $\langle D,w,\#\rangle$

$\Rightarrow$ $\langle M,w\rangle \in A_{TM} \Rightarrow M(w)$ accetta $\Rightarrow \langle M,w,a\rangle \in W_{TM}$, $\langle M,w,a\rangle = f(\langle M,w\rangle)$

$\Leftarrow$ $\langle M,w\rangle \notin A_{TM} \Rightarrow M(w)$ rifiuta $\Rightarrow \langle D,w,\#\rangle \notin W_{TM}$, $\langle D,w,\#\rangle = f(\langle M,w\rangle)$

la riduzione è corretta $\Rightarrow$ W_{TM} indecidibile

⇒) Linguaggi Turing-ric. voglio mostrare che ∃ MTM M che lo riconosce,

Sia N la TM che lo riconosce, io posso sempre simulare una MTM con una TM a singolo nastro

usando # ∉ Σ ∉ Γ per delimitare i vari nastri

$$\# w_1 \ldots w_n \# \sqcup \# \sqcup \# \ldots \# \sqcup \#$$

e marcando le variabili su cui stanno le varie testine

↓

$$\# \dot{w}_1 \ldots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \ldots \# \dot{\sqcup} \# \qquad \Delta$$

⇐) So che una MTM riconosce L voglio mostrare che sia Turing-riconoscibile da una generica TM

Questo è ovvio perché posso usare una MTM come una TM usando solo un nastro $\square$

3 Complessità — 10 Points

- Mostrare che il seguente problema, detto 1/2-*CLIQUE*, è *NP*-completo: Dato un grafo G con un numero pari di vertici, stabilire se esiste una *clique* in G di taglia uguale alla metà del numero di vertici nel grafo.
- Definire il problema 2-*SAT* e mostrare che esso appartiene a P.

RIDUCO $\frac{1}{2}$-CLIQUE A 3SAT

3SAT $\leq_m^P$ $\frac{1}{2}$ CLIQUE

$f: \Sigma^* \rightarrow \Sigma^*$

SIA P LA TM CHE CALCOLA f

F: CONVERTE $\langle \phi \rangle \in$ 3SAT IN UN GRAFO PER CLIQUE

- INTRODUCE 3K NODI DOVE K = NUMERO CLAUSOLE IN ϕ
- OVVIAMENTE I NODI x E $\bar{x}$ NON HANNO ARCHI
- TUTTI GLI ALTRI NODI SONO COLLEGATI
- AGGIUNGO ALTRI K NODI COLLEGATI CON AGLI ALTRI

(IN TOTALE AGGIUNTO 4K NODI E 2K-CLIQUE)

- PER DEFINIRLO RIDUCIAMO 2-SAT AL PROBLEMA DELLA RICERCA DELLE COMP FORTEMENTE CONNESSE

CREIAMO IL GRAFO G CHE HA I VERTICI LE VARIABILI DI ϕ ($\phi(x_1 \dots x_n)$) E HA ARCHI TRA NODI SE:

$$(x_1 \vee x_2) \in \phi \Rightarrow (x_1, x_2) \in E(G)$$

$$(x_1 \vee \bar{x}_2) \in \phi \Rightarrow (\bar{x}_1, \bar{x}_2) \in E(G)$$

LEMMA 1) SE HO UNA COMP-FORT CONNESSA HO SIA x CHE $\bar{x}$ ϕ NON È SAT

POICHÉ SE AVESSI $x \dots x_i \dots \bar{x} \dots x_j$

AVREI $x \to x_i \to \bar{x} \to x_j$

MA QUINDI ANCHE $x \to \bar{x}$ E OVVIAMENTE $\bar{x} \to x$

$x \to \bar{x} = \bar{x} \vee \bar{x} = \bar{x}$

$\bar{x} \to x = x \vee x = x$

RISULTA CHIARO CHE ϕ NON SAT

LEMMA 2) SE x E $\bar{x}$ NON NELLA STESSA COMP $\Rightarrow$ ϕ È SAT

FACCIO UN ORDINE TOPOLOGICO NEL GRAFO

PER UNA VARIABILE x HO 2 CASI

- $x \in C_i$ e $\bar{x} \in C_j$ con $i < j \Rightarrow x = 0$
- $x \in C_i$ e $\bar{x} \in C_j$ e $i > j \Rightarrow x = 1$

A QUESTO PUNTO NON $\exists$ ARCHI (a,b) CON $a=1$ E $b=0$

AFFERMIAMO PER ASSURDO CHE

$a=1$ e $b=0$ con $(a,b) \in E(G)$

$\exists$ $(\bar{b},\bar{a})$ con $\bar{b}=1$ e $\bar{a}=0$

poiché $b=0$ $\bar{b} \in C_k$ con $k>i>j$

$$C_j \to C_i \to C_k$$

$\bar{a}$ è quindi a C_l con $l<k$

$$C_j \to C_i \to C_l \to C_k$$

ma poiché ho arco $(\bar{b},\bar{a})$ l'ordine topologico è violato

NON $\exists$ ARCHI DI TIPO $1 \to 0$ $\Rightarrow$ SAT GARANTITA