Grammatiche

Es Scrivere una grammatica (a potenza minima) che generi

$$L = \{ab^{n_1}ab^{n_2}\dots ab^{n_k} \mid k \geq 2;\ \forall 1 \leq i \leq k \text{ si ha } n_i > 0;\ \exists\, \underline{i,j},\ i \neq j \text{ t.c. } n_i = n_j\}$$

non devono essere unici

esempi di stringhe di L:

abab

$aba\underbrace{bb}abbbabbbba\underbrace{bb}$ $k=5,\ i=2,\ j=5$

...

ababab

...

Con un ASF non si può fare, con un APND sì: cerco una grammatica context-free.

Idea: prima genero i due blocchi di 'b' aventi la stessa lunghezza e poi, prima, dopo e tra di loro produco altri blocchi del tipo ab^+

Inizio a generare stringhe del tipo

$$X\, a b^n\, Y\, a b^n\, Z$$

e poi X, Y, Z produrranno delle stringhe di $(ab^+)^*$.

Possiamo fare così   genera ab^nYab^n servono per generare $(ab^+)^+$ prima o dopo ab^nYab^n

$$S \longrightarrow B \mid AB \mid BA \mid ABA$$

$$B \longrightarrow aY$$

$$Y \longrightarrow bYb \mid bCb$$

$$C \longrightarrow a \mid Aa$$

$$A \rightarrow aD \mid aDA$$

$$D \longrightarrow bD \mid b$$

come funziona?

per generare $abab^2ab^3ab^2$:

$$S \longrightarrow AB \longrightarrow AaY \longrightarrow AabYb$$
$$\rightarrow AabbCbb \rightarrow Aab^2Aab^2$$
$$\rightarrow Aab^2aDab^2 \rightarrow Aab^2abDab^2$$
$$\rightarrow Aab^2abbDab^2 \rightarrow Aab^2ab^3ab^2$$
$$\rightarrow aDab^2ab^3ab^2 \rightarrow abab^2ab^3ab^2.$$

<u>Es</u> Trovare una grammatica per

$$L = \{ a^{(2^n)} \mid n \geq 0 \} = \{ a, a^2, a^4, a^8, a^{16}, \dots \}$$

Un APND non basta: ci vuole una MT, quindi una grammatica di tipo 0 o 1.

Idea: genero n simboli (che servono per i raddoppi) ed <u>una</u> 'a': per ognuno degli n simboli faccio in modo di raddoppiare le 'a'.

$S \longrightarrow a \mid XAY$

$X \longrightarrow XC$ il numero delle 'C' sarà n

$CA \longrightarrow AAC$

come funziona? per generare $a^8 = a^{(2^3)}$:

$CY \longrightarrow Y$

$A \longrightarrow a$

$X \longrightarrow \varepsilon$

$Y \longrightarrow \varepsilon$

$S \rightarrow XAY \xrightarrow{3} XCCCAY \rightarrow XC\boxed{CA}ACY$
$\rightarrow XCAAC\boxed{AC}Y \rightarrow X\boxed{CA}AAACCY$
$\rightarrow XAACAAAC^2Y \rightarrow XA^4\boxed{CA}AC^2Y$
$\rightarrow XA^6\boxed{CA}C^2Y \rightarrow XA^8C^3Y \xrightarrow{\cdots} a^8$

Che tipo di grammatica è? <u>tipo 0</u>

Può esistere una grammatica di tipo 1 per L?

Sì, un automa lineare può riconoscere L,

quindi c'è una grammatica di tipo 1 per L,

ma è piuttosto complicata:

$S \longrightarrow aa \mid aaaa \mid BXAE \mid a$

$XA \longrightarrow AAX$

$XE \longrightarrow YAAE \mid ZAAA$

$AY \longrightarrow YA$

$BY \longrightarrow BXA$

$AZ \longrightarrow ZA$

$\boxed{BZ \longrightarrow AAA}$

$A \longrightarrow a$

così NON uso $E \to \varepsilon$

serve per cancellare B evitando di avere $B \to \varepsilon$

X, Y e Z "simulano" un cursore che scorre la stringa da sx a dx (X) o da dx a sx (Y e Z).

X serve a raddoppiare le A, Y e Z a "riavvolgere" il nastro, Z indica l'ultimo "riavvolgimento" che fa sparire B ed E

provate a farla funzionare

<u>Es</u> Descrivere un APND che riconosca il complementare del linguaggio

$$L = \{ w \cdot w \mid w \in \{a,b\}^* \}$$ sull'alfabeto $\{a,b\}$.

$L = \{ \varepsilon, aa, bb, aaaa, abab, baba, bbbb, aaaaaa, \ldots \}$

è il linguaggio dei "quadrati perfetti".

Che tipo di linguaggio è? C'è un APND per L? NO

w "al contrario"

NB: Il ling. $\{ w \cdot w^R \mid w \in \{a,b\}^* \}$ è riconosciuto da un APND.

Se L non è riconosciuto da un APND, è lecito dire che neppure il complementare è ricon. da un APND? NO: ci sono linguaggi riconosciuti da AP il cui complementare non è riconosciuto da un AP; questo è il caso.

Passiamo a L^c (complementare di L):

come sono fatte le sue stringhe.

Ci sono 3 casi.

- se una stringa ha lunghezza <u>dispari</u> di sicuro <u>non</u> è del tipo $w \cdot w$, quindi sta in L^c.

- se una stringa u ha lunghezza pari, sta in L^c se nella prima metà c'è una 'a' e nella seconda una 'b' posizionata rispetto al centro di u come 'a' rispetto all'inizio (o viceversa: prima 'b' e poi 'a').

quindi u si può "scomporre" così

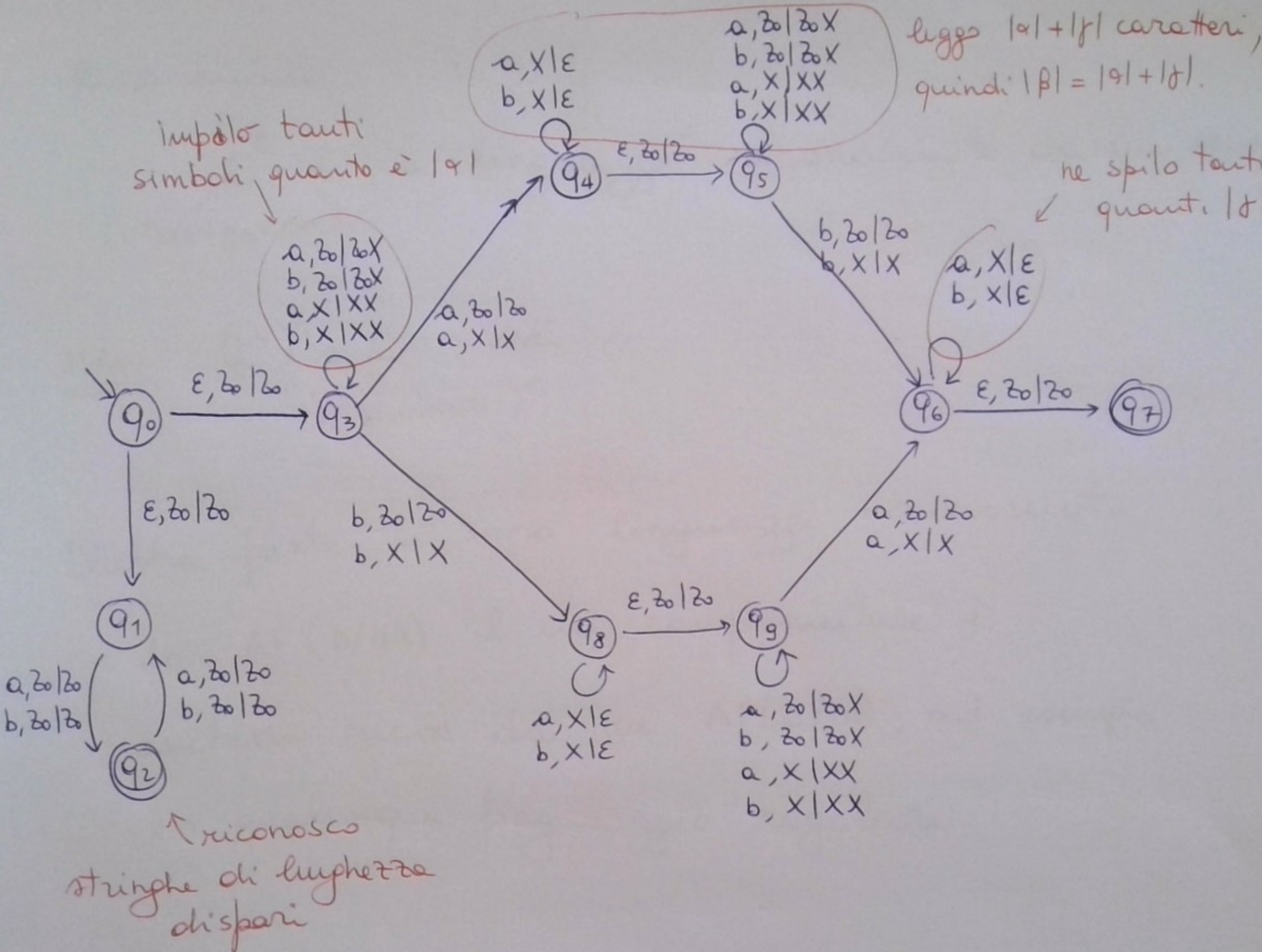
$$u = \underline{\alpha}\, a\, \beta\, b\, \gamma \qquad \text{con } \alpha, \beta, \gamma \in \{a,b\}^*$$
$$\text{e } |\alpha| + |\gamma| = |\beta|$$

$$u = \underbrace{ab}_{\alpha} \cdot a \underbrace{ab\,ab}_{\beta} b \cdot \underbrace{ab}_{\gamma}$$

$$u = \underbrace{ab}_{\alpha} \cdot a \underbrace{abbb\,ab}_{\beta} b \cdot \underbrace{abbb}_{\gamma}$$

dunque

$$L^c = \{ w \mid |w| \text{ dispari} \} \cup$$
$$\{ w \mid \exists \alpha, \beta, \gamma \text{ tc } w = \alpha a \beta b \gamma \text{ con } |\alpha\gamma| = |\beta| \} \cup$$
$$\{ w \mid \exists \alpha, \beta, \gamma \text{ tc } w = \alpha b \beta a \gamma \text{ con } |\alpha\gamma| = |\beta| \}.$$



impilo tanti simboli quanto è |α|
leggo |α|+|γ| caratteri, quindi |β| = |α|+|γ|.
ne spilo tanti quanti |γ|
riconosco stringhe di lunghezza dispari
q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
ε, Z0|Z0
a, Z0|Z0X
b, Z0|Z0X
a, X|XX
b, X|XX
a, Z0|Z0
a, X|X
b, Z0|Z0
b, X|X
a, X|ε
b, X|ε
a, Z0|Z0
b, Z0|Z0

Riassumendo:

L^c è un linguaggio riconosciuto da un APND

(i "non-quadrati")

MA $(L^c)^c = L$ no

(i "quadrati")

D'altra parte ci sono linguaggi riconosciuti da AP (D/ND) il cui complementare è anch'esso ricon. da un AP(D/ND), ad esempio un qualunque linguaggio regolare.

Modelli descrittivi: logica (MFO/FSO) sulle parole

Sono logiche interpretate: dominio, lettere funzionali, predicati e costanti sono fissati e hanno un significato ben determinato.

Il senso di una formula di MFO/MSO è determinato una volta fissata una stringa su un alfabeto dato.

no $w = \varepsilon$

Ad esempio, se $\Sigma = \{a, b\}$ e $w \in \{a, b\}^+$

e F è una f.b.f. di MFO/MSO, allora

F va interpretato sul dominio

$\{0, 1, \dots, |w| - 1\}$ ← sono le posizioni delle lettere in w, che partono dalla 0.

le variabili del primo ordine sono naturali da 0 a $|w| - 1$, quelle del secondo sono sottoinsiemi del dominio.

Abbiamo come predicati: $=, <, \leq, >, \geq$ (binari)

e i predicati "monadici" $a(x)$ e $b(x)$, dove

$a(x)$ è vero sse nella posizione x di w c'è la lettera 'a'.

Le costanti 'a' e 'b', la funzione successore ($x \mapsto x+1$)
Volendo si possono avere altri predicati/funzioni, purché definibili da quelli di base (ad es, last(x)...)

es. Sia $\Sigma = \{a, b\}$, descrivere il linguaggio formato dalle stringhe di Σ^+ che verificano le seguenti formule.

① $\forall x\,(\,\mathrm{last}(x))$.
w verifica la formula sse ogni posizione di w è l'ultima: possibile sse w ha un'unica posizione, cioè lunghezza 1.

Il linguaggio definito dalla formula $\{a, b\}$.

② $\underbrace{\neg \exists x \exists y \, (x<y \wedge a(x) \wedge b(y))} \wedge \forall x \, (b(x) \Rightarrow \exists y \, (x<y \wedge a(y)))$

conviene "studiare" separatamente
le due sottoformule e
poi fare l'<u>intersezione</u> dei
linguaggi che si trovano

Inizio da $\neg \exists x \exists y \, (x<y \wedge a(x) \wedge b(y))$

equivale a $\forall x \forall y \, (\neg(x<y) \vee \neg a(x) \vee \neg b(y))$

o anche $\forall x \forall y \, (x<y \Rightarrow \neg(a(x) \wedge b(y)))$

o anche $\forall x \forall y \, ((a(x) \wedge b(y)) \Rightarrow x \geq y)$

w: ... b ... a ... ok
$y \leq x$

... a ... b ... non ok
$x < y$

Queste formule definiscono le stringhe
che dopo una 'a' non hanno delle 'b',

quindi $b^* a^* \setminus \varepsilon$

ovvero $b^* a^+ + b^+ a^*$ ovviamente regolare:

le formula data è di MFO,
quindi definisce un ling. regolare,
anzi un regolare "star-free"

Poi : $\forall x (b(x) \rightarrow \exists y (x < y \wedge a(y)))$

"se in posizione x c'è 'b', allora esiste una posizione
successiva (un po' dopo la x) in cui c'è 'a' "

sono tutte le stringhe che terminano con 'a' ovvero $\Sigma^* \cdot a$.

In conclusione, la formula originaria definisce $b^* a^+$.

es. Sia $\Sigma = \{a, b\}$, trovare una formula di MFO/MSO (scegliendo sempre la logica "a potenza minima") per i linguaggi

① $aa^*b^* = a^+b^*$

② insieme delle stringhe con 'a' in ogni posizione dispari (ad esempio $\underset{0\ 1\ 2\ 3\ 4\ 5}{b\,a\,b\,a\,b\,a}$ posiz.) va bene anche aaaa

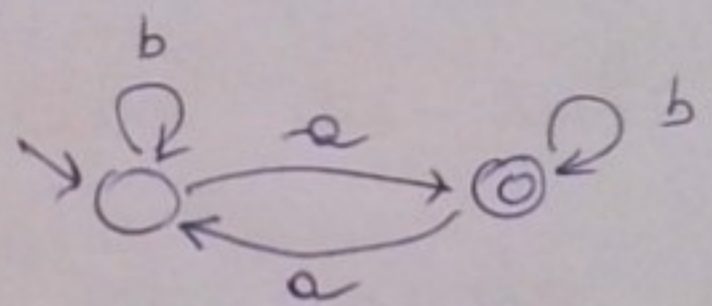
③ insieme delle stringhe in cui compare un numero dispari di 'a'.

<u>Oss</u> 1, 2, 3 si possono fare con MFO/MSO?

Vediamo se sono regolari: 1 di sicuro lo è, 2? 3?

2 è regolare: $((\Sigma a)^* \cancel{a} + (\Sigma a)^* \Sigma) \setminus \varepsilon$

anche 3 è regolare: facile con un ASF



a^+b^*

. Per 1, basta MFO: è simile al precedente (basta scambiare 'a' con 'b'), ci sono anche formule più semplici come

$$a(0) \wedge \forall x \forall y \left((a(x) \wedge b(y)) \Rightarrow x < y \right).$$

• Per 2, sicuramente si può fare con MSO
(si può dimostrare, ma non sapete come,
che MFO non basta).

Uso quantificazioni del secondo ordine:
quantificherò esistenzialmente un
predicato unario (cioè un sottoinsieme
di posizioni) e farò in modo che risulti
vero sse l'argomento è una posizione
dispari e imporrò che in tali posizioni
ci sia 'a'.

$$y = x - 2$$

$$\exists P \Big(\neg P(0) \wedge \forall x \forall y \big((P(x) \wedge x = y+2) \Rightarrow P(y)\big) \wedge \forall x \,(P(x) \Leftrightarrow \neg P(succ(x)))$$

$$\wedge \; \forall z \,(P(z) \Rightarrow a(z)) \Big).$$

P è l'insieme di tutte (e sole) le posizioni dispari di w

Significato: esiste un insieme di posizioni

tc. non contiene 0 e se contiene un

~~posiz~~ numero x contiene $x-2, x-4, x-6, \dots$

tc. ogni lettera di w che occupa una

posizione di P è una 'a'.

Ad esempio, se $w = bababa$, allora il dominio

è $\{0,1,2,3,4,5\}$, P sarà $\{1,3,5\}$