GRAMMATICA NON AMBIGUA PER ESPRESSIONI

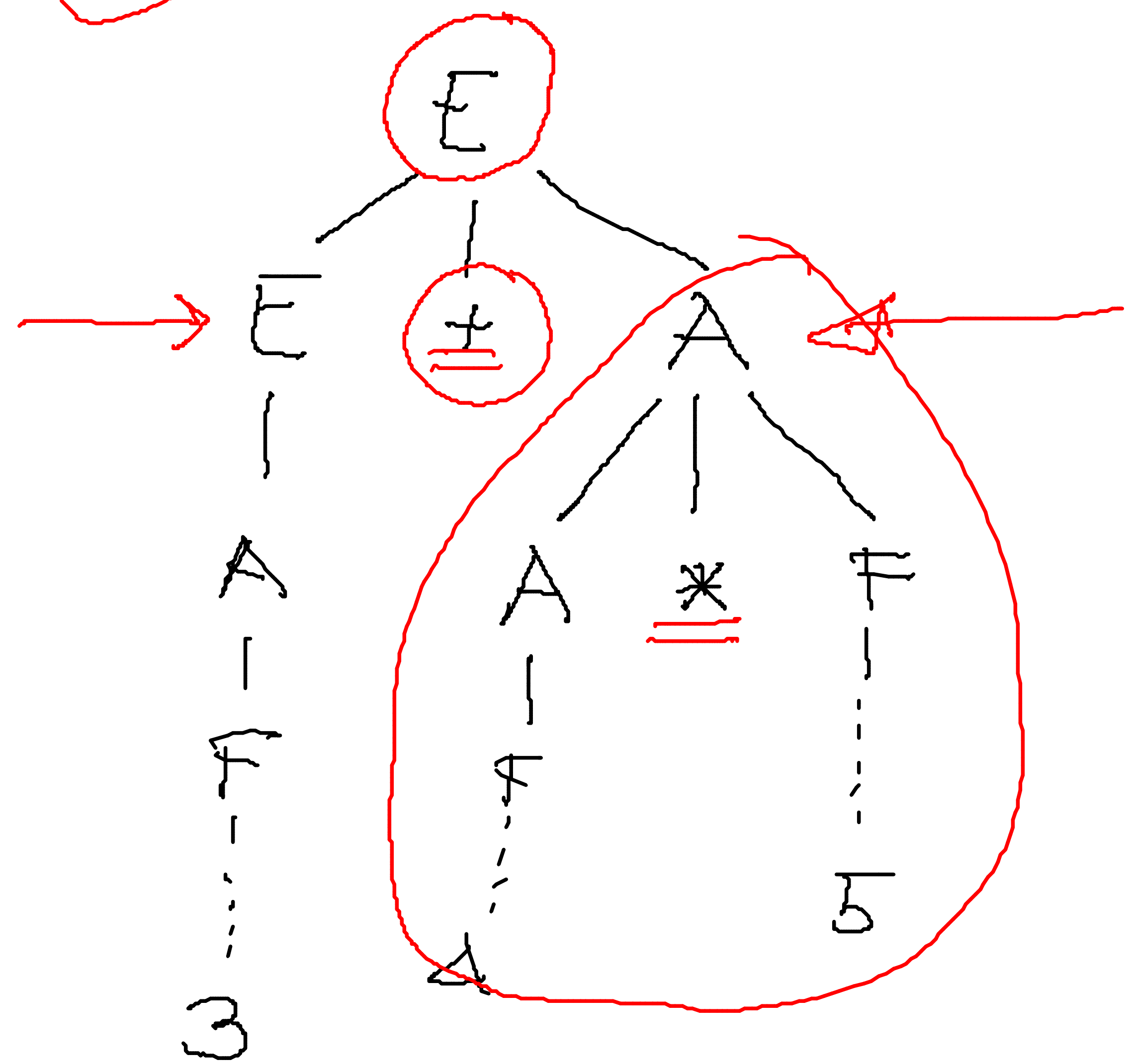
$E \to E + A \mid E - A \mid A$

$A \to A * F \mid A / F \mid F$

$F \to (E) \mid N$

$3 + 4 * 5$

23



$$L = \{ a^{2n} b^{n} \mid n \geq 1 \} \qquad \Lambda = \{a, b\}$$

$$\underbrace{a \cdots\cdots a}_{2n}\ \underbrace{b \cdots\cdots b}_{n}$$

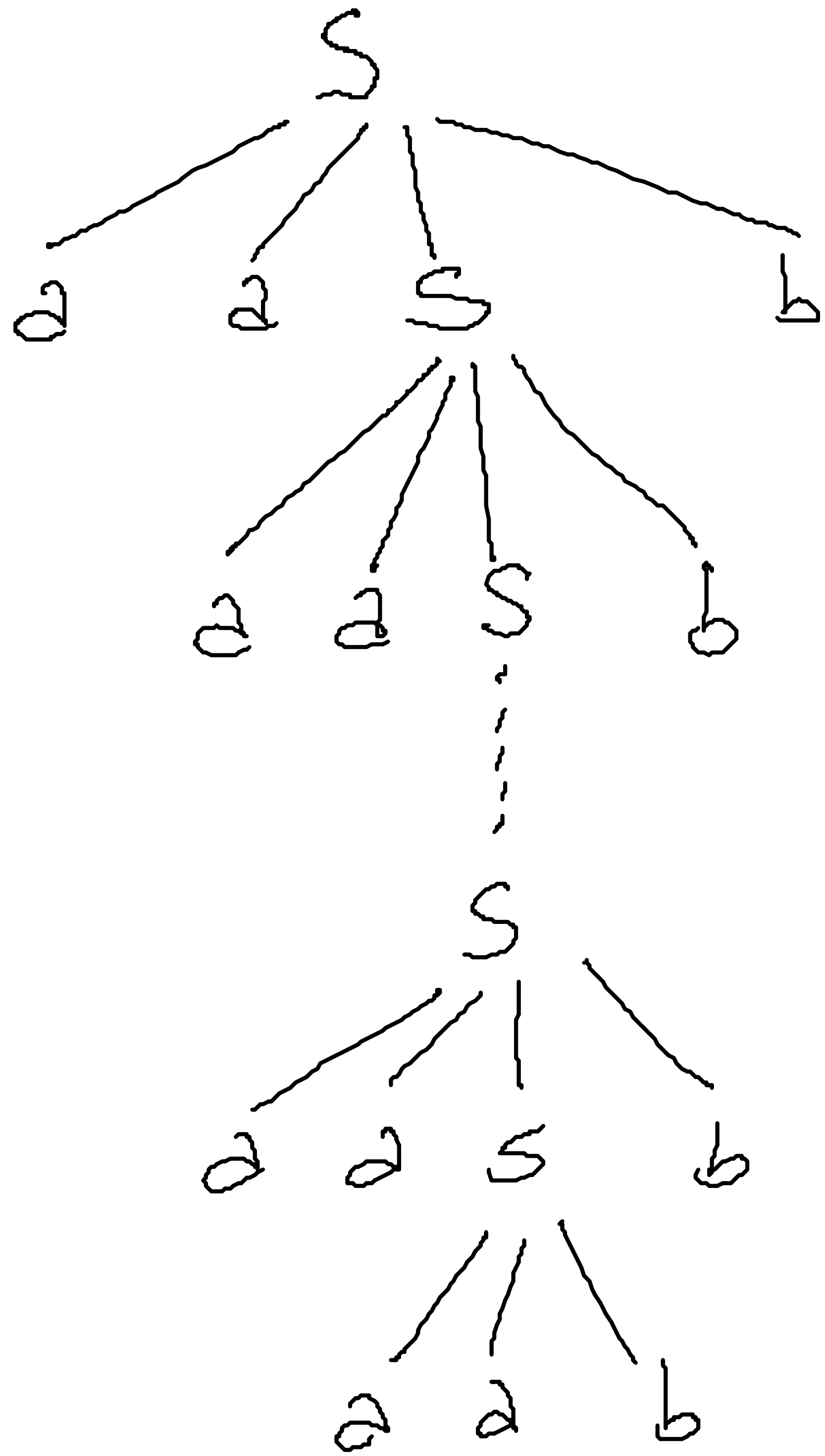
$aab \in L$

$aaaabb \in L$

$aabb \notin L$

$S \to aaSb \mid aab$

$S \to aaSb \mid aab$



$$\mathcal{L} = \{ a^n c^k b^{n+1} \mid n \geq \emptyset, \ k \geq 1 \} \qquad \Lambda = \{a, b, c\}$$

$$\underbrace{a \cdots a}_{n} \ \underbrace{c \cdots c}_{k} \ \underbrace{b \cdots b}_{n+1}$$

$cb \in \mathcal{L}$ $\qquad n = 0 \quad k = 1$

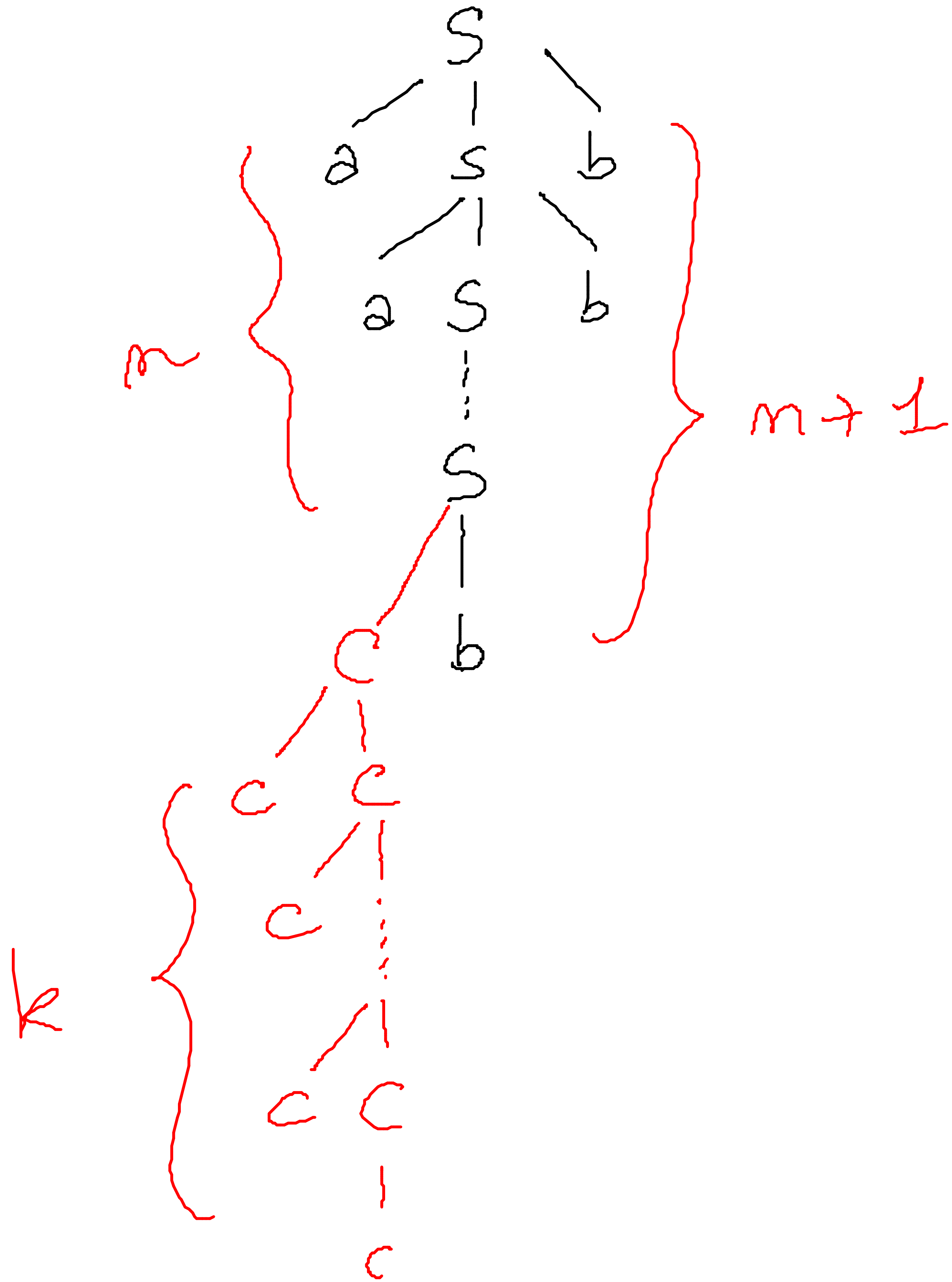
$acccccbb \in \mathcal{L}$ $\qquad n = 1 \quad k = 4$

$acb \notin \mathcal{L}$

$S \to aSb \mid b$ $\qquad \mathcal{L}' = \{ a^n b^{n+1} \mid n \geq 0 \}$

$S \to aSb \mid Cb$

$C \to cC \mid c$



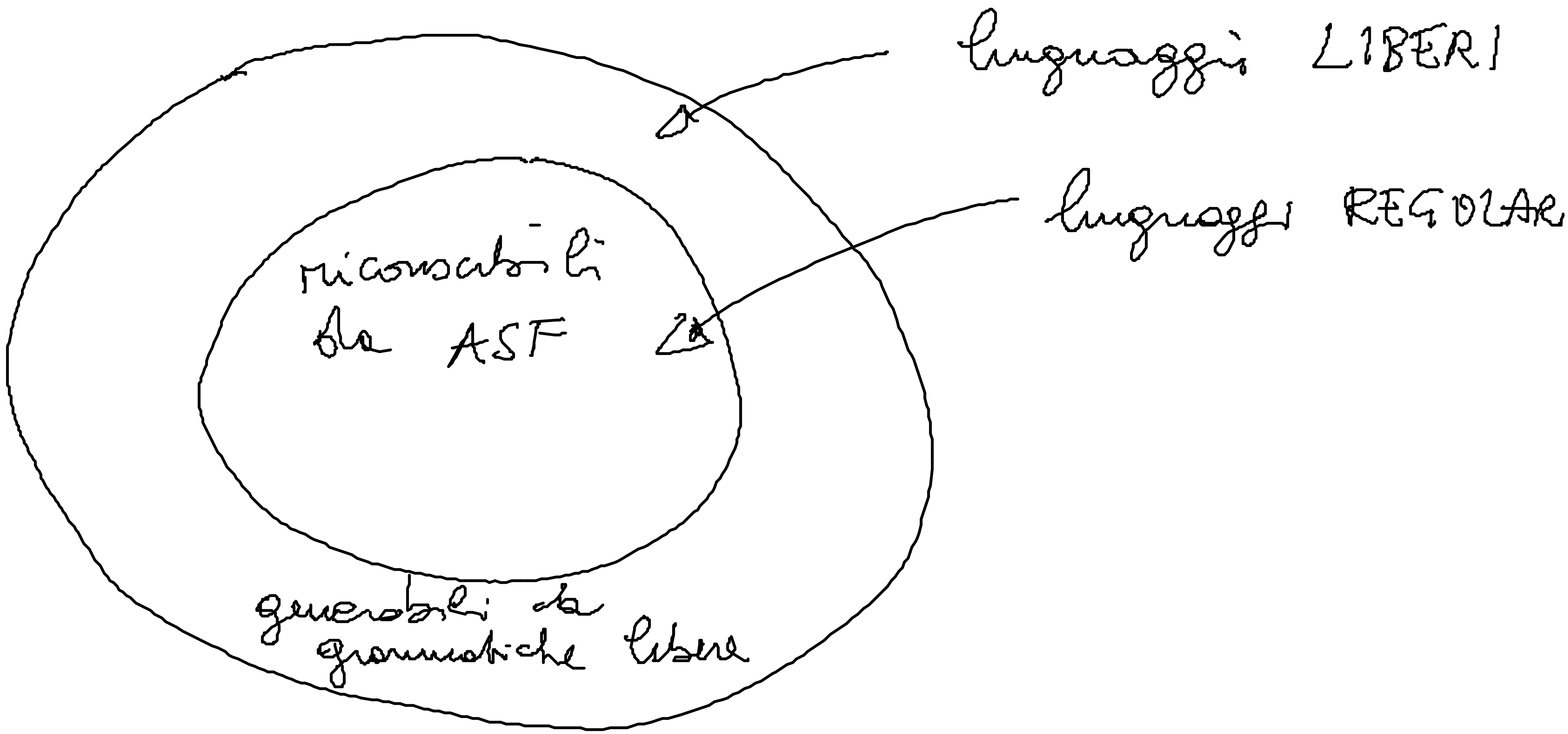
$$\mathcal{L} = \{ a^n b^n c^n \mid n \geq 1 \} \qquad \Lambda = \{a, b, c\}$$

non è libero; non esiste una grammatica libera da contesto che genera il linguaggio

Relazione tra ASF e GRAMMATICHE LIBERE

- le grammatiche libere sono più "potenti" (da un punto di vista espressivo) degli automi

1) se L è un linguaggio riconoscibile da un automa ASF, allora L è generabile da una grammatica libera
2) esiste almeno un linguaggio generabile da una grammatica libera ma non riconoscibile da un automa


linguaggi LIBERI
linguaggi REGOLARI
riconoscibili da ASF
generabili da grammatiche libere

DIMOSTRAZIONE: se L è riconoscibile da ASF A allora L è generabile da una G

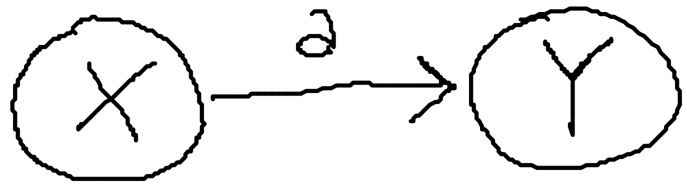
Per COSTRUZIONE - Sia A l'automa dato

$$A = \langle \Lambda, \Sigma, F, I, \delta \rangle$$

$$G_A = \langle \Lambda, V_A, S_A, P_A \rangle$$

- per ogni STATO $X \in \Sigma$ introduciamo una categoria sintattica X in V_A
- S_A è la categoria sintattica corrispondente a I

- per ogni $\langle \langle X, a \rangle, Y \rangle \in \delta$



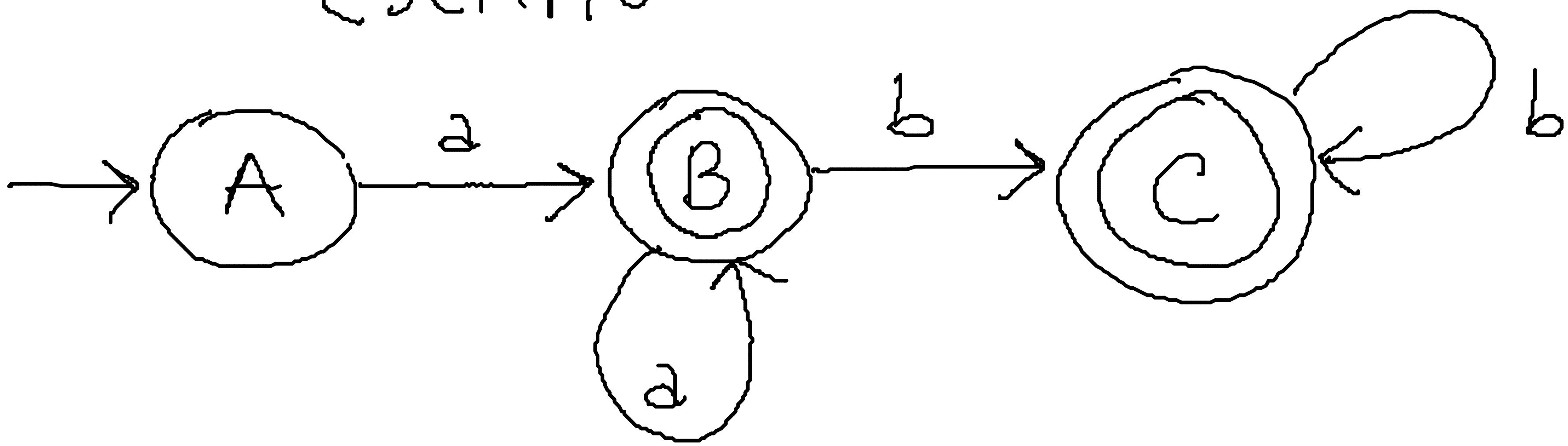
introduciamo la produzione

$$X \to \underline{\underline{aY}} \quad \text{in } P$$

se $Y \in F$ introduciamo anche la produzione

$$X \to \underline{\underline{a}}$$

ESEMPIO

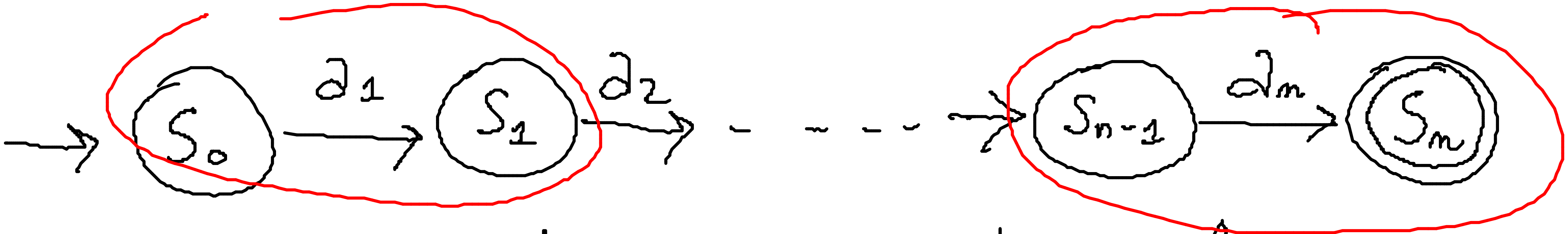


$$\mathcal{L} = \{ a^n b^k \mid n \geq 1,\ k \geq 0 \}$$

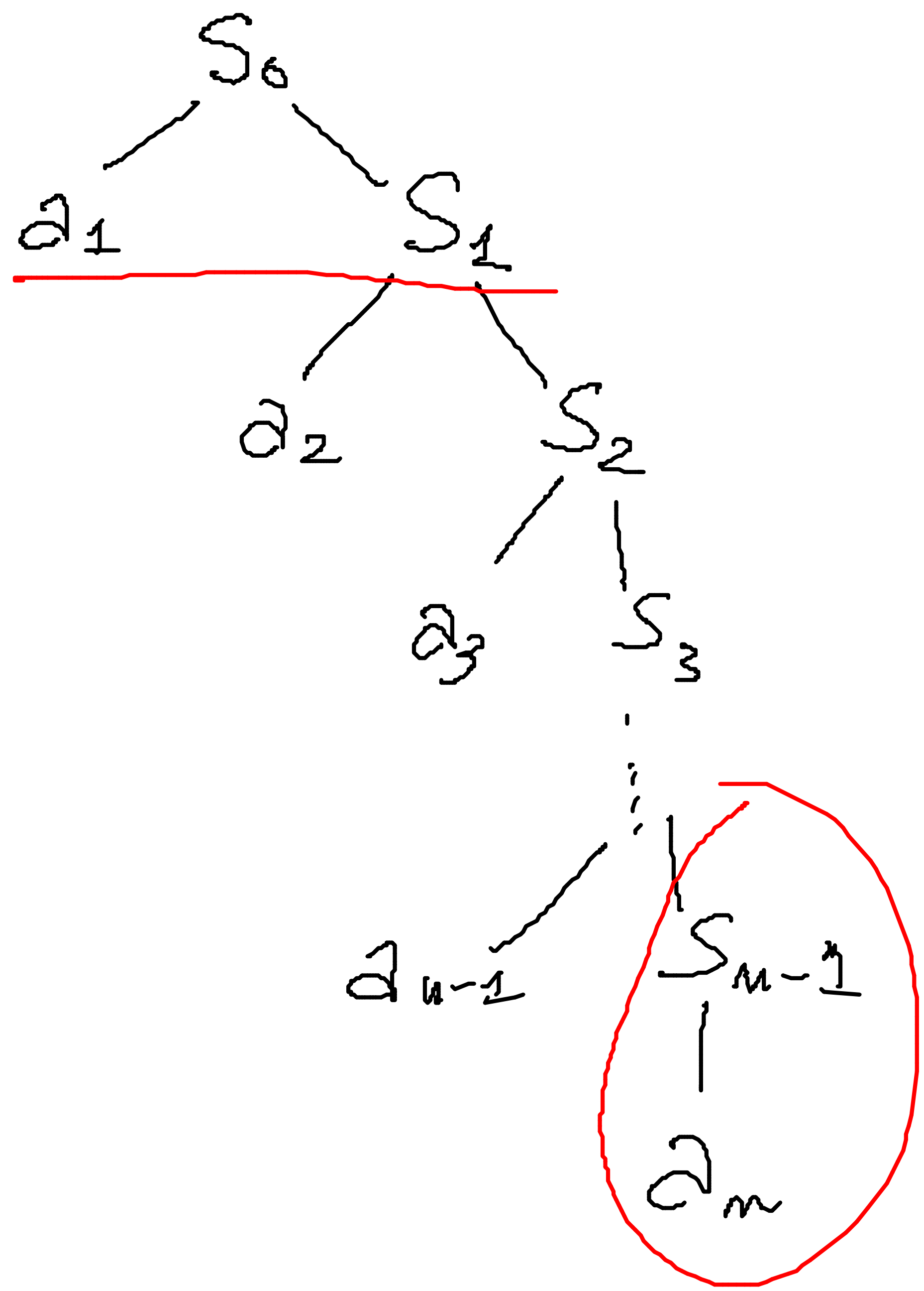
$$A \to aB \mid a$$

$$B \to aB \mid a \mid bC \mid b$$

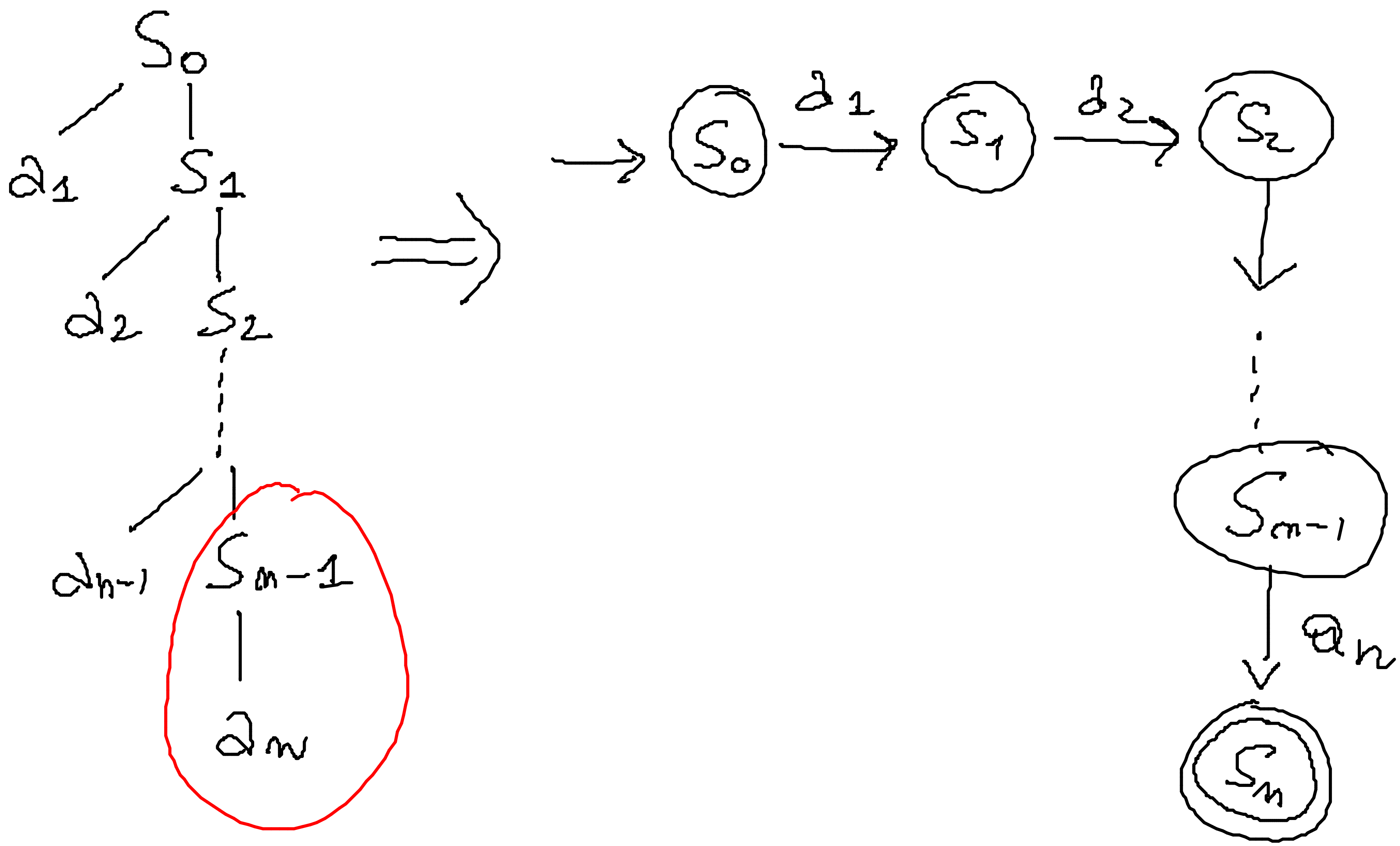
$$C \to bC \mid b$$



cammino di riconoscimento in A della stringa $a_1 \dots a_n$



Sia $a_1 \dots a_n$ frontiera di un albero di derivazione in G_A



GRAMMATICHE REGOLARI

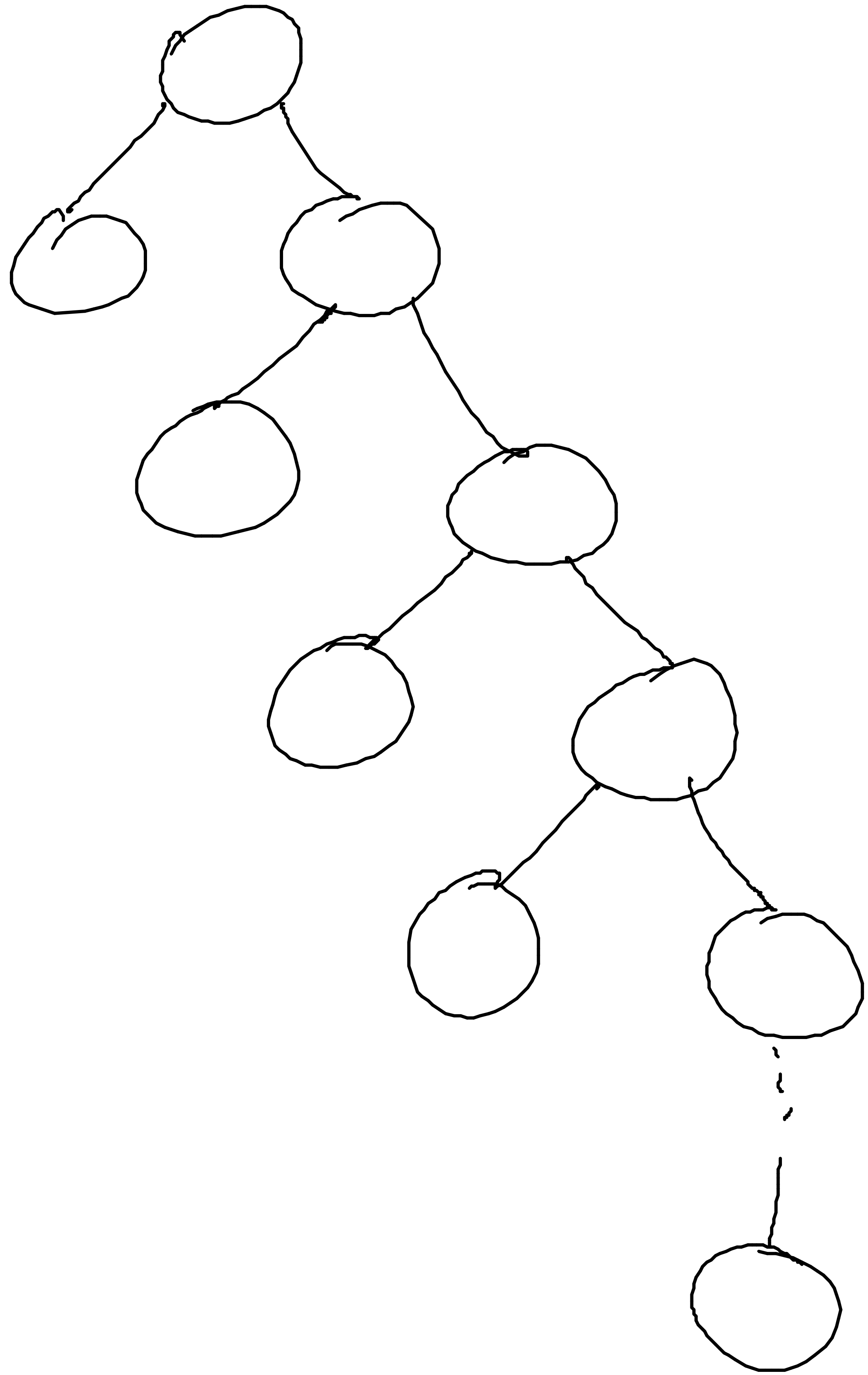
Una grammatica libera si dice REGOLARE se le produzioni sono della forma

$$A \to X_1 X_2 \qquad \text{con } X_1 \in \Lambda$$
$$X_2 \in V$$

oppure

$$A \to X_1 \qquad \text{con } X_1 \in \Lambda$$

grammatiche regolari SINISTRE



- Data una grammatica libera G, esiste un automa "equivalente"? (cioè che riconosce $\mathcal{L}(G)$?

In generale NO!

Teorema Non esiste un ASF che riconosce

$$\mathcal{L} = \{ a^n b^n \mid n > 0 \}$$

Dim. per assurdo

Supponiamo (per assurdo) che tale ASF esista

Applichiamo il cosiddetto

PRINCIPIO DELLE BUCHE DEI PICCIONI

(pigeon hole principle)

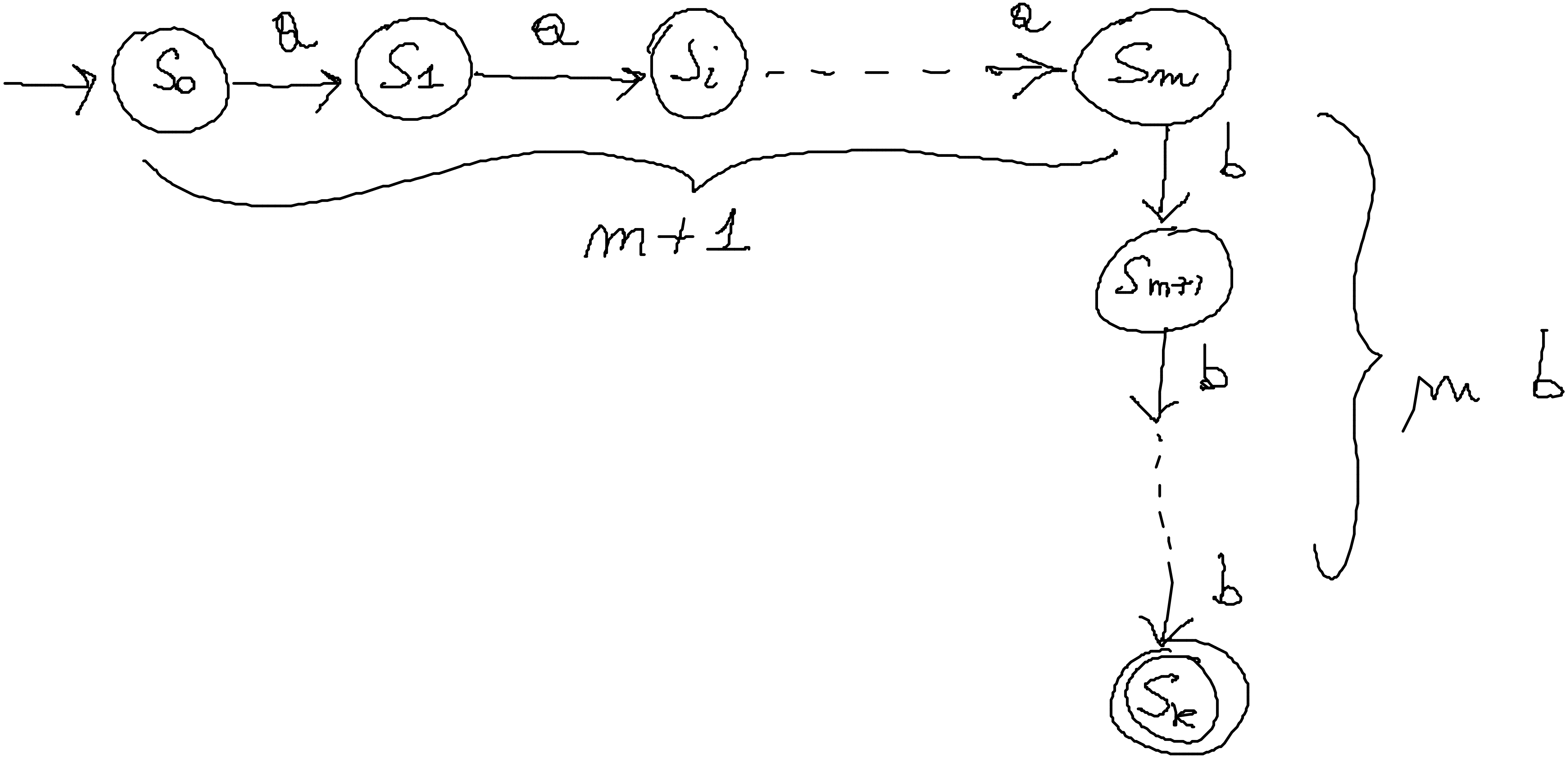
Se ci sono $n+1$ piccioni e n buche,

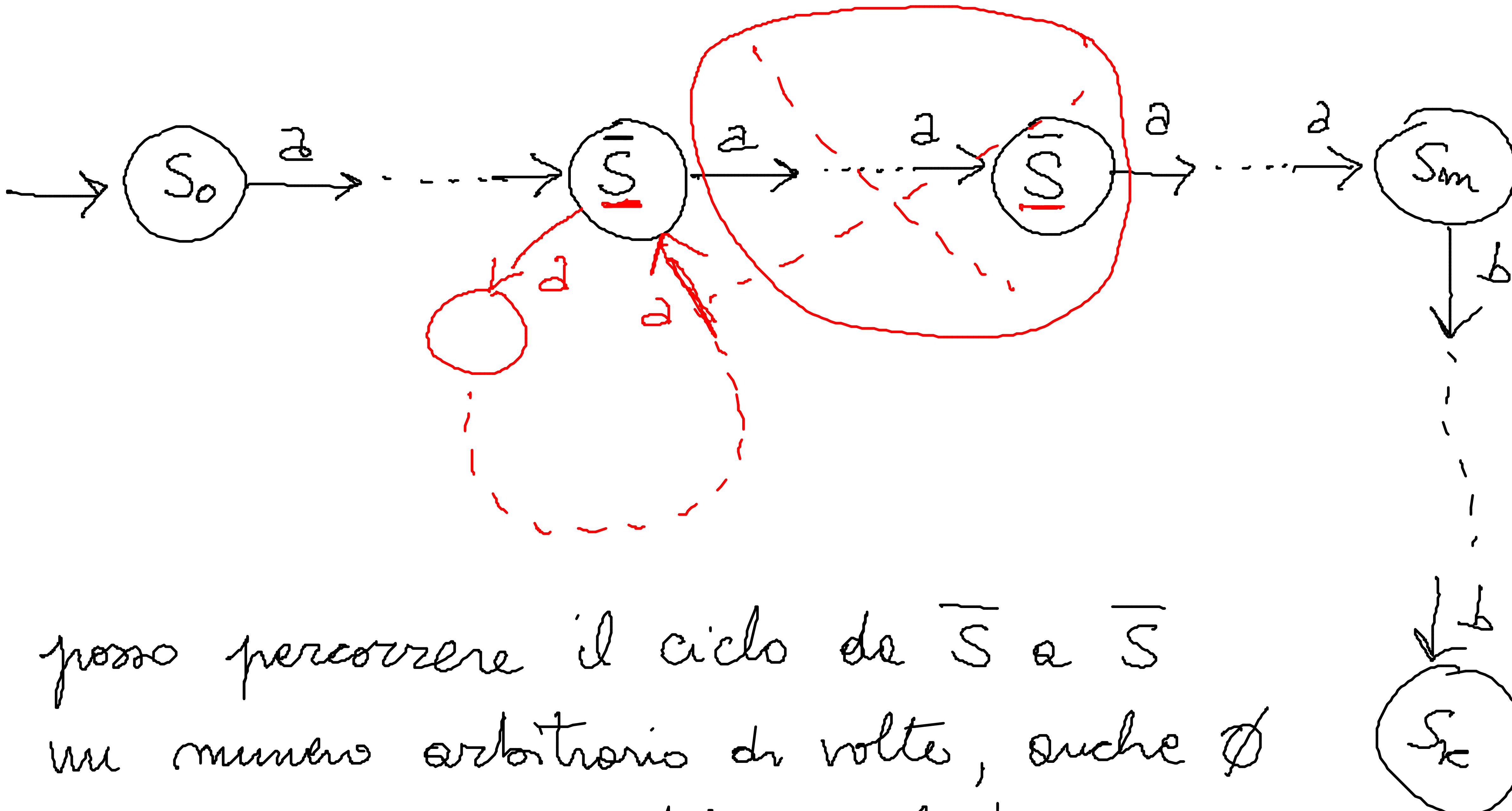
2 piccioni devono finire nella stessa buca

Supponiamo che l'ASF abbia $\underline{\underline{n}}$ stati

Consideriamo la sequenza

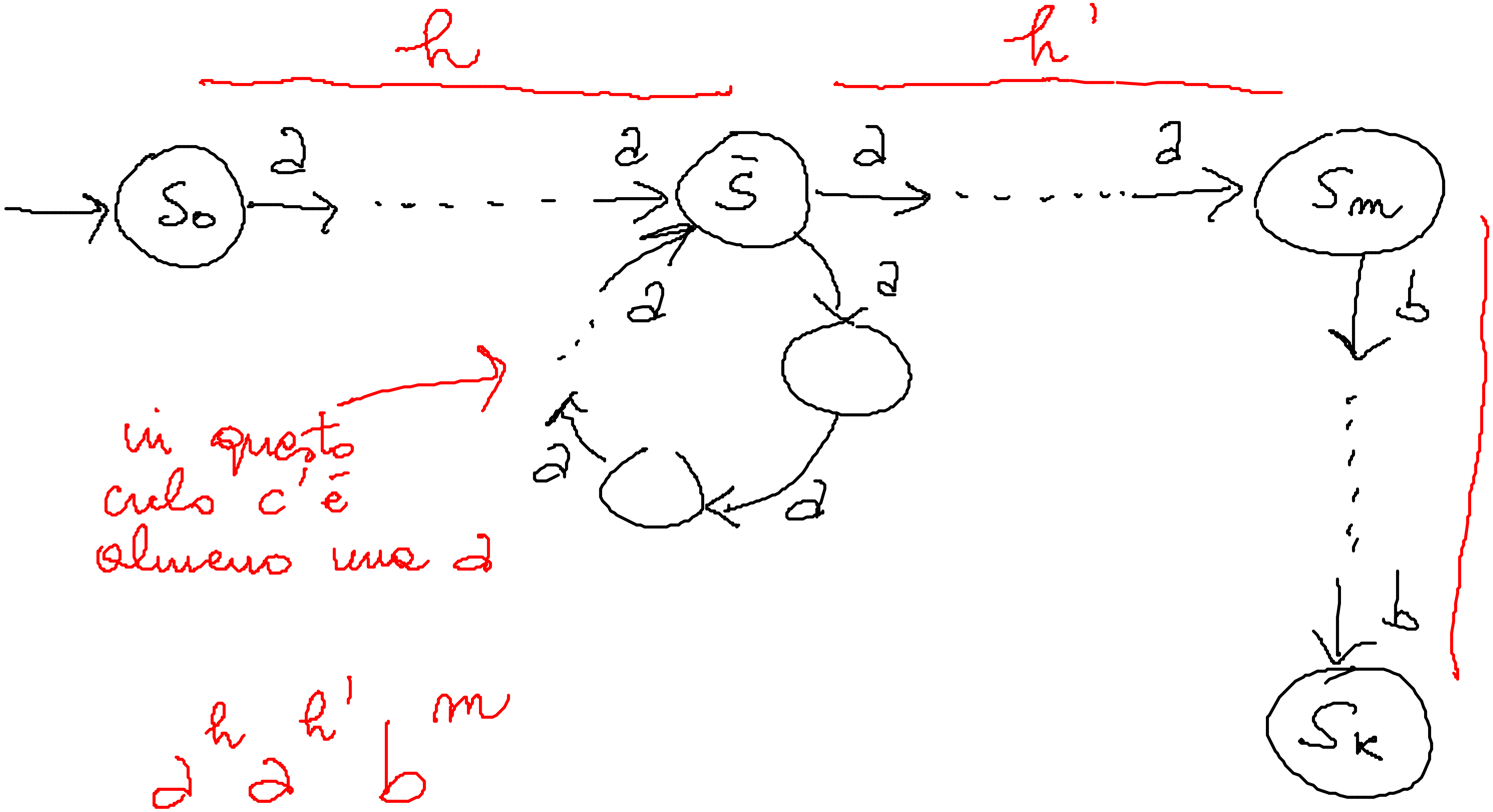
$$a^m b^m \in L \quad \text{con} \quad m > n$$





posso percorrere il ciclo da $\bar{S}$ a $\bar{S}$ un numero arbitrario di volte, anche $\emptyset$ e riconoscere una stringa del tipo

$$a^k b^m \quad \text{con } k \neq m$$



$$a^h a^{h'} b^m$$

$$h + h' < m$$

Assurdo! ASF riconosce $\{a^n b^n \mid n > 0\}$ e ho trovato $a^x b^y$ riconosciuta dall'automa con $x \neq y$ q.e.d. c.v.d.

PUMPING LEMMA

Per ogni linguaggio REGOLARE L esiste una costante n tale che, per ogni stringa $z \in L$ con $|z| > n$, esiste una scomposizione

$$z = uvw \quad \text{con}$$

- $|uv| \leq n$
- $v \neq \varepsilon$
- per ogni $i \geq 0$ anche $uv^iw \in L$

$|z|$ è la lunghezza di z

$|z| > n$

$z = u\,v\,w$ con $v \neq \varepsilon$

$\underline{u\,w} \in L$

$u\,v\,v\,w \in L$

$u\,v\,v\,v\,w \in L$

$\vdots$

$u\,\underbrace{v \ldots\ldots v}_{i \text{ volte}}\,w \in L$

DIMOSTRAZIONE

- se L è un linguaggio finito il lemma è banale.

Supponiamo che L contenga stringhe la cui lunghezza massima è k

scelgo $n > k$ e il lemma è dimostrato

$$k = \max \{ \ell \mid \ell = |\alpha|, \alpha \in L \}$$

$$n > k$$

per ogni $z \in L$ con $|z| > n$ ......

parentesi

"per ogni $x \in A$ vale $p(x)$"

è vero se per ciascun $a \in A$ vale $p(a)$

è falso se per qualche $a \in A$ non vale $p(a)$

Supponiamo A = insieme vuoto.

Vale

$$\forall x \in A.\ p(x)$$

è vero se A è vuoto

qualunque sia p

L non è finito, contiene un numero infinito di stringhe

Poiché L è regolare è riconoscibile da un ASF

Sia A un automa con N stati che riconosce L

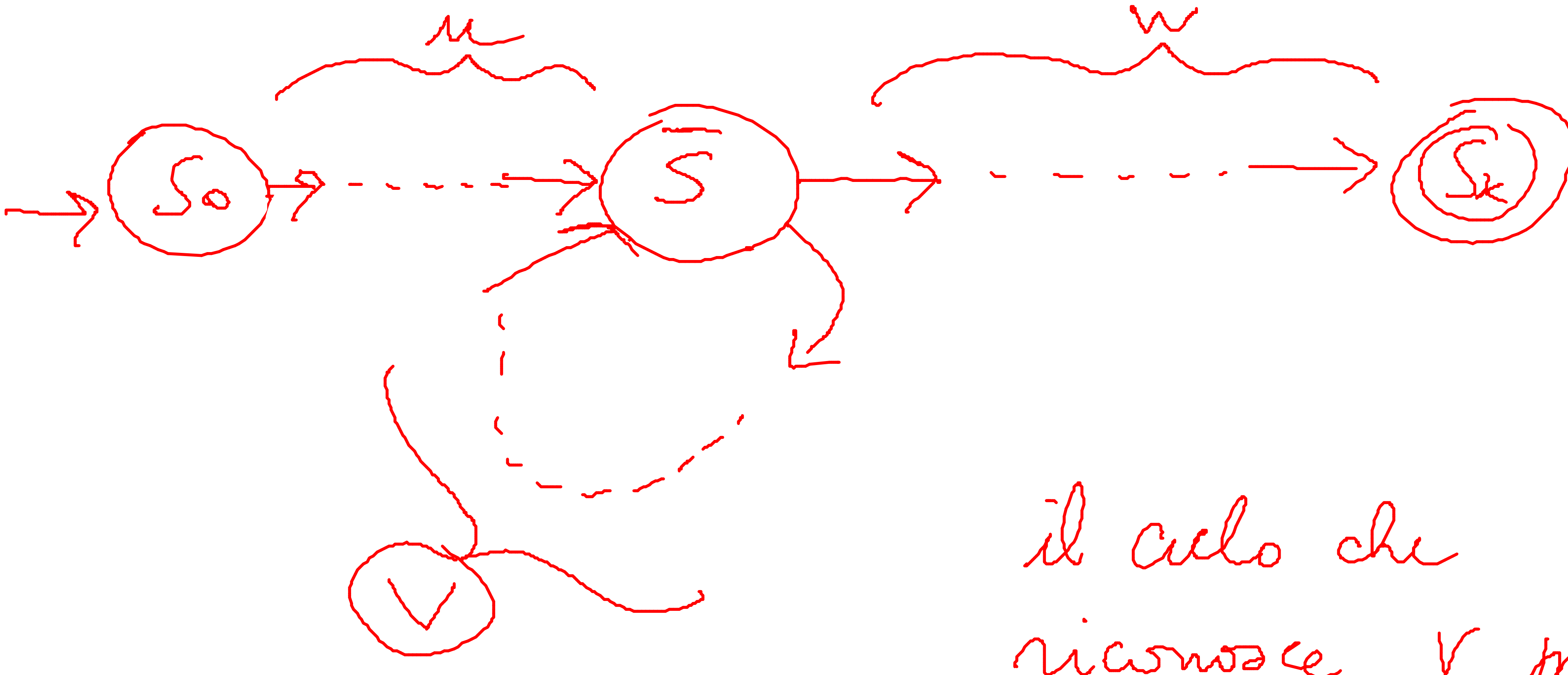
Sia z una qualunque stringa $z \in L$ con $|z| > N$

Per il pigeon hole principle esiste un cammino

$$\underbrace{S_0, \ldots, \bar{S}}_{u}, \underbrace{\ldots, \bar{S}}_{v}, \underbrace{\ldots, S_k}_{w}$$

con S_0 stato iniziale, S_k è uno stato di riconoscimento e il cammino riconosce z

- indichiamo con U la sequenza riconosciuta da S_0 alla prima occorrenza di $\bar{S}$
- indichiamo con V la sequenza riconosciuta nel ciclo da $\bar{S}$ a $\bar{S}$
- indichiamo con W la sequenza riconosciuta dalla seconda occorrenza di $\bar{S}$ fino a S_k



il ciclo che riconosce v può essere percorso i volte

q.e.d.