TEO (PUMPING LEMMA PER CFL) Sia L un CFL. Allora esiste p tale che se $w \in L$ con $|w| \geq p$ è possibile scomporre $w = uvxyz$ in modo che:

(i) $\forall i \geq 0$; $uv^ixy^iz \in L$

(ii) $|vy| > 0$

(iii) $|vxy| \leq p$.

DIM. Wlog assumiamo G in forma normale di Chomsky. Pertanto l'albero di derivazione sarà binario.

AFF. Ogni albero generato da G il cui cammino più lungo ha lunghezza i, genera stringhe di lunghezza $\leq 2^{i-1}$.

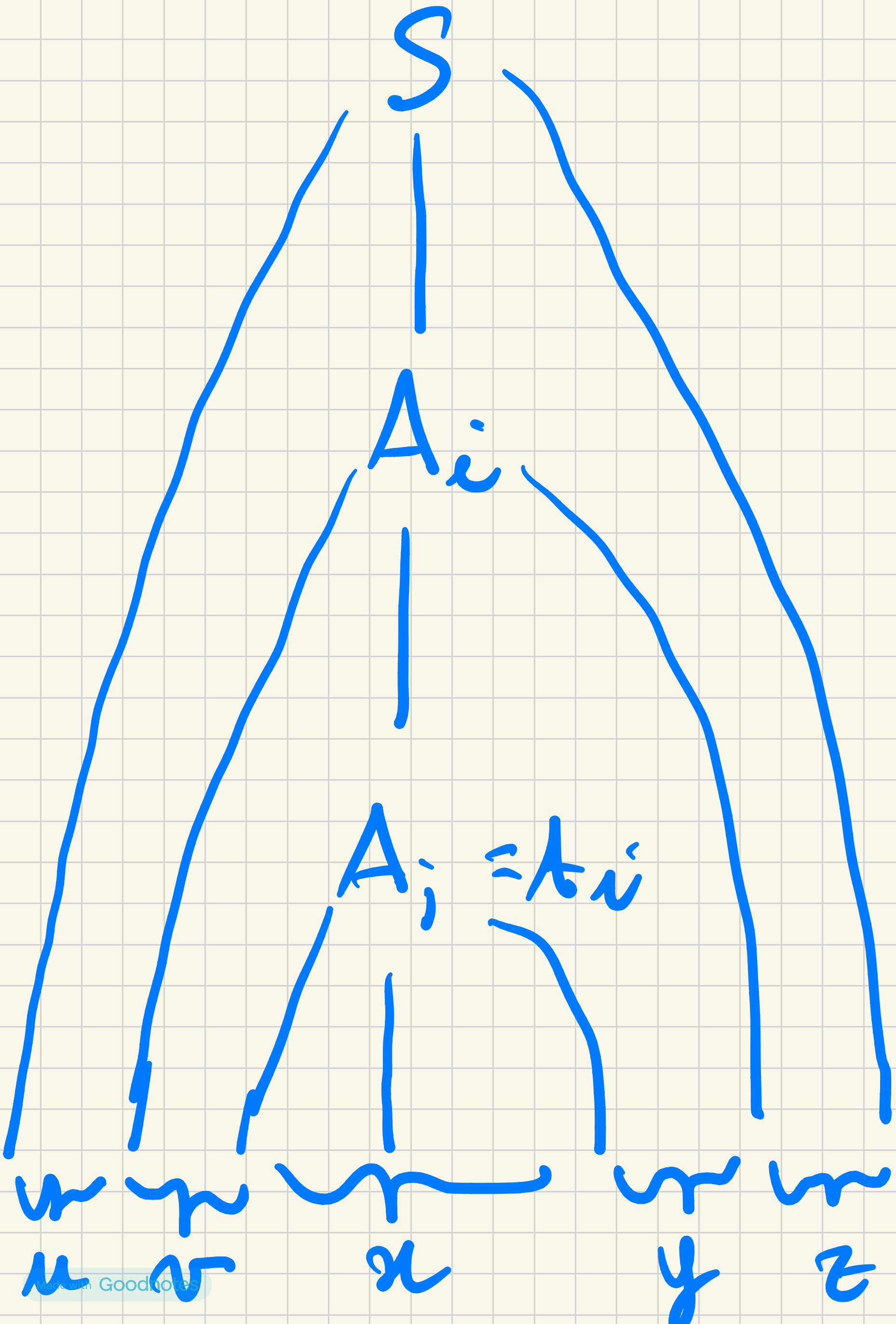
DIM. Per induzione sulla lunghezza del cammino. Se il cammino è lungo 1 abbiamo $S \to a$ dove a è un terminale e infatti $|a| = 2^{1-1} = 2^0 = 1$.

Sia vero per cammini di lunghezza $i-1$ e supponiamo il cammino più lungo sia di lunghezza $i > 1$. La prima sostituzione sarà $S \to AB$ perché G è in forma normale e il cammino è lungo almeno 2. I sottoalberi radicati in A e B hanno lunghezza $i-1$ e per induzione generano stringhe lunghe al più $2^{i-1-1} = 2^{i-2}$. Pertanto S genera una stringa lunga $\leq 2 \cdot 2^{i-2} = 2^{i-1}$ $\blacksquare$

Made with Goodnotes

Sia m la $\# V$ in G. Pongo $p = 2^m$.
Segue che una stringa w t.c. $|w| \geq p$
è generata da un cammino lungo $\geq m+1$.
Questo cammino ha $m+2$ nodi di
cui uno è terminale e ci sono $m+1$
variabili. Conseguenza: almeno una
variabile si deve ripetere.
Scegliamo la prima ripetizione nella
sequenza lunga $m+1$, siano le variabili

$A_i = A_j$ con $i, j \le m+1$, $i \ne j$.



L'albero di radice A_i genera vxy, mentre l'albero di radice $A_j = A_i$ genera x. Le stringhe u e z sono la parte terminale sx e dx del cammino

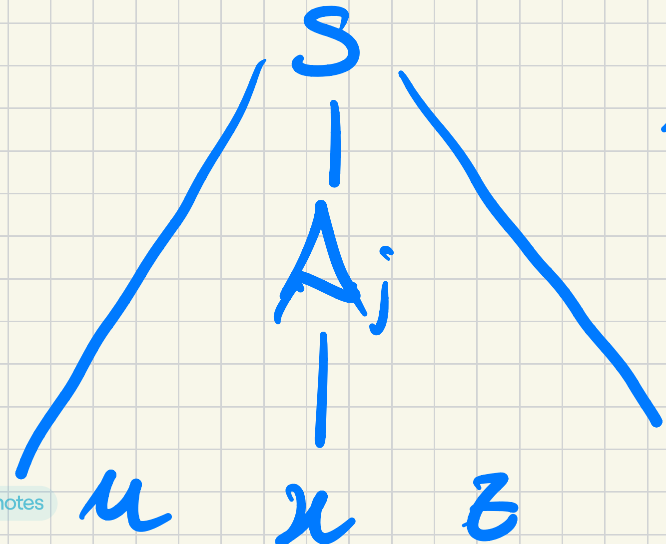
$$S \leadsto A_i$$

Restano da controllare le condizioni del Teorema:

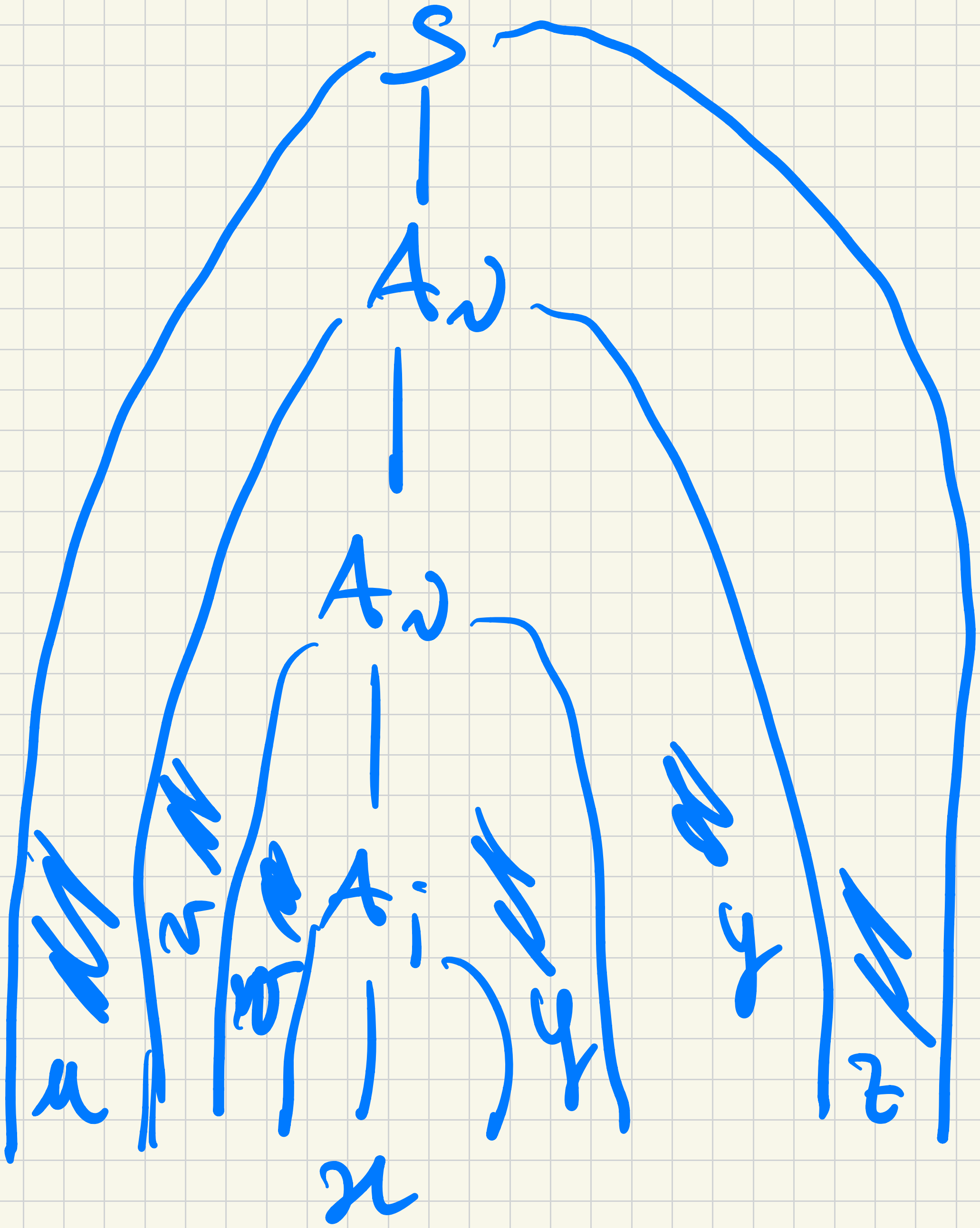
(ii) $vy \neq \varepsilon$. Questo perché G consente di derivare $A_i \overset{(*)}{\Rightarrow} A_j$. Ma le uniche regole compatibili con questo in G sono del tipo $A_i \to BC$ t.c. C deve derivare A_j e B derivare v oppure $y \Rightarrow vxy \neq x \Rightarrow |vy| > 0$.

(iii) $|vxy| \leq p$. Questo perché abbiamo scelto A_i come la prima ripetizione in una sequenza lunga $\le m+1$. Ovvero questo cammino genera stringhe di lunghezza $\le 2^{m+1-1} = 2^m = p$.

(ii) Posso sostituire A_i con A_j e viceversa!



$u v^0 x y^0 z \in L$



$$uv^2xy^2z \in L$$

$$\Longrightarrow \quad \forall \nu \geq 0$$

$$uv^\nu xy^\nu z \in L$$

$\blacksquare$

Esempio: $L = \{0^n 1^n 2^n : n \geq 1\}$ non è CFL.

Applico il PUMPING LEMMA. Suppongo lo sia e prendo $w = 0^p 1^p 2^p$ dove p è il valore del pumping. Ovviamente $|w| = 3p > p$. Sia $w = uvxyz$; siccome $|vxy| \leq p$, abbiamo che vxy contiene al più 2 simboli tra '0', '1', '2'.

$0 \cdots \underbrace{0\, 1 \cdots}\, 1\ 2 \cdots 2$ Ad es.

Se w sono (in vxy) '0' e '1' non ci sono '2'!

Se $i = 0$ e considero:

$u v^0 x y^0 z$ e siccome $vy \neq \varepsilon$

sto rimuovendo due simbolo senza modificare il numero di occorrenze del terzo simbolo: $u v^0 x y^0 z \notin L$.

MACCHINE DI TURING

Abbiamo visto diversi modelli di calcolo, principalmente DFA/NFA e PDA. Sappiamo anche che ci sono linguaggi non riconoscibili da questi automi.
Le TM sono state introdotte da Turing nel 1936 e corrispondono al modello astratto di computer.
Vedremo che ci sono problemi intrinseca=

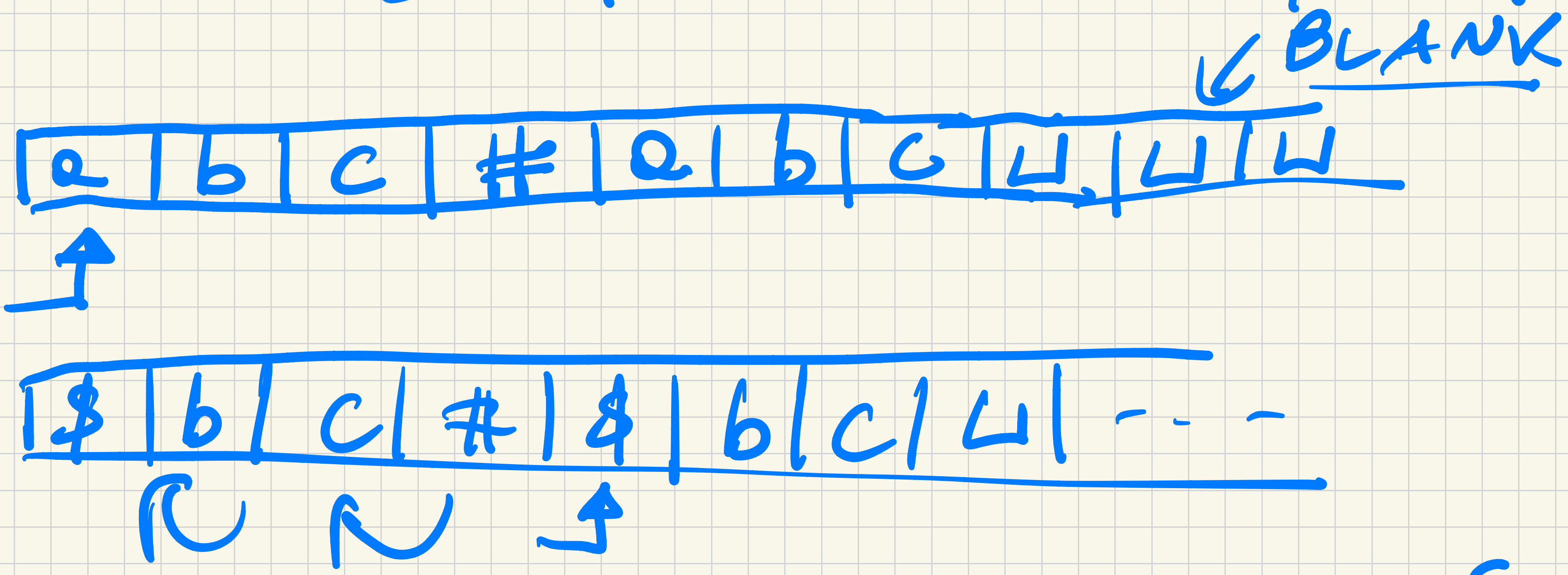
mente non risolubili dai computer indipendentemente dalle risorse.

A livello intuitivo, una TM ha un nastro di lunghezza infinita; questo nastro contiene input. La TM può leggere / scrivere sul nastro e muovere la testina a SX o DX.

Inoltre c'è stato di accettazione / rifiuto con effetto immediato.

Esempio: Una TM per

$$L = \{ w\#w : w \in \{0,1\}^* \}$$



muove a sx e ripete.

DEF (TM) Una TM è una tupla

$$M = (Q, \Sigma, \Gamma, \delta, q_{start}, q_{acc}, q_{rej})$$

dove:

- Q è insieme finito di stati.
- Σ è insieme finito di input ($\sqcup \notin \Sigma$)
- Γ è insieme finito simboli di nastro ($\sqcup \in \Gamma$, $\Sigma \subseteq \Gamma$)

LEFT RIGHT

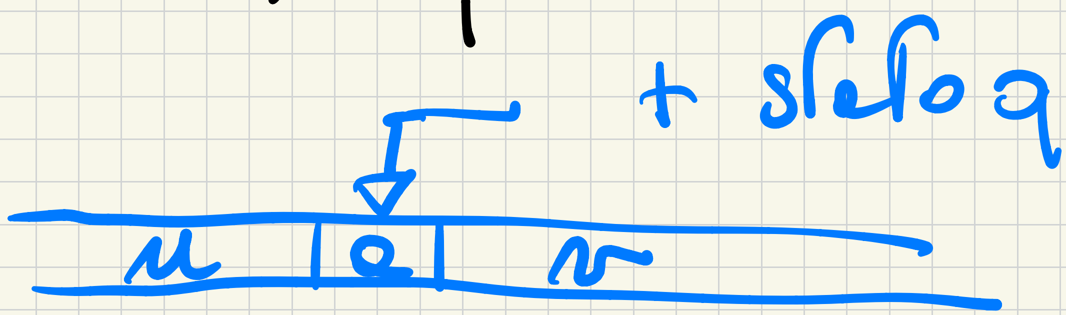
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$

- $q_{start}, q_{acc}, q_{rej} \in Q$ e sono gli stati iniziale, di accettazione e di rifiuto ($q_{rej} \neq q_{acc}$).

Considerare le configurazioni: stato, contenuto nastro e posizione testina.
Per convenzione si usa la notazione

$$u\,q\,a\,v \qquad u, v \in \Gamma^*, \quad q \in Q, \quad a \in \Gamma$$

$\Rightarrow$



Configurazione iniziale: $q_{start}\, w$
dove w è l'input.
Configurazione accettazione/rifiuto: lo stato $q \in \{q_{acc}, q_{rej}\}$.

$$u\, a\, q_i\, b\, v \vdash_M u\, q_j\, a\, c\, v$$

sse $\delta(q_i, b) = (q_j, c, L)$

similmente

$$u\, a\, q_i\, b\, v \vdash_M u\, a\, c\, q_j\, v$$

sse $\delta(q_i, b) = (q_j, c, R)$

Abbiamo che M accetta $w \in \Sigma^*$ sse $C_{start} = q_{start}\, w$ è t.c.

$$C_{start} \vdash_M^* C_{acc}$$

dove C_{acc} è una qualsiasi configurazione di accettazione.

DEF (LINGUAGGI TURING RICONOSCIBILI).

Un linguaggio è Turing riconoscibile se $\exists$ TM M che lo riconosce, ovvero $\forall\ w \in L$ abbiamo che M accetta w.

Attenzione: Se $w \notin L$ $M(w)$ può rifiutare o andare in loop.

Decisore: Non va mai in Loop, ovvero: $w \in L \Rightarrow M$ accetta w

$w \notin L \Rightarrow M$ rifiuta w

DEF (DECIDIBILITÀ) L è Turing DECIDIBILE se ∃ TM M che la DECIDE.