

QUIZ SU GERARCHIA DI CHOMSKY

Forse: **Nessun linguaggio** (tranne forse i linguaggi finiti) è chiuso rispetto all'operazione di subset

Grammatiche di tipo 0 / struttura di frase:

- Producono **linguaggi ricorsivamente numerabili** (semi-decidibili) che sono chiusi rispetto a :
 - Concatenazione
 - Unione
 - Kleene *
 - intersezione
- Producono linguaggi riconosciuti dalle MdT

Grammatiche di tipo 1 / dipendenti dal contesto / monotone:

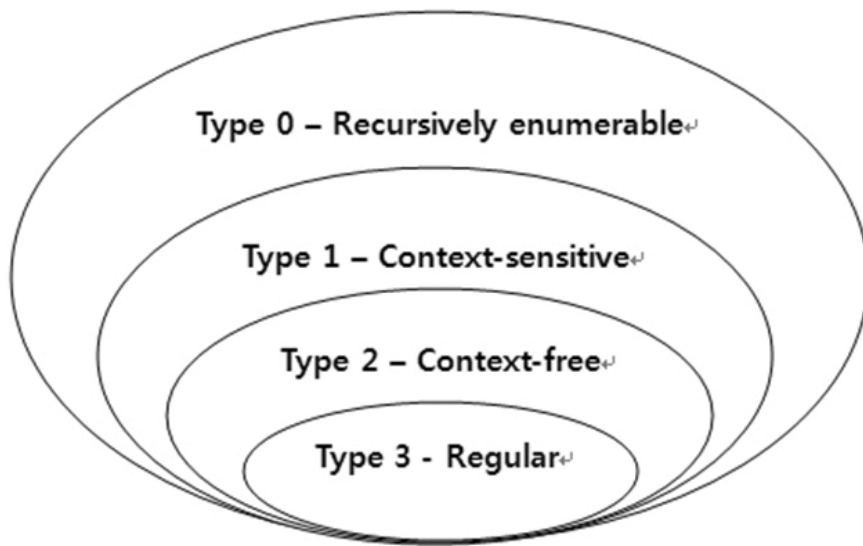
- Producono **linguaggi ricorsivi** (decidibili) che sono chiusi rispetto a :
 - Complemento
 - Concatenazione
 - Unione
 - Kleene *
 - Intersezione
 - Differenza

Grammatiche di tipo 2 / libere dal contesto:

- Crea **linguaggi liberi dal contesto** (no $\varepsilon \rightarrow$ f. nor. chomsky e greibach) che sono chiusi rispetto a :
 - Concatenazione
 - Unione
 - Kleene *
- Producono linguaggi riconosciuti dagli automi a pila
- Il suo **complemento** è **decidibile**

Grammatiche di tipo 3 / regolari:

- Queste grammatiche **coincidono con** l'unione delle **grammatiche lineari** destre e sinistre
- Producono **linguaggi regolari** le cui closure props sono:
 - Complemento
 - Concatenazione
 - Kleene *
 - Unione
 - Differenza
 - Intersezione
- I **linguaggi finiti** fanno parte di questo linguaggi (tutto ciò che non è un linguaggio finito dovrebbe essere un linguaggio infinito). Chiusi rispetto a:
 - Concatenazione
 - Unione
 - Differenza
 - Intersezione
 - La chiusura di Kleene produce un linguaggio regolare
 - Il complemento produce un linguaggio regolare
- Producono linguaggi riconosciuti da NFA / DFA / δ NFA (che sono equivalenti si arrestano dopo un numero finito di transizioni se ricevono una sequenza finita di simboli) e possono essere rappresentati come una regexpr
- Forse: Un sottoinsieme di un linguaggio finito è regolare



QUIZ SULLE CARDINALITÀ

Se arriva una cardinalità semplice ragionarci!

- Cardinalità delle stringhe su Σ con $n > 0$ simboli: Σ^n
- Numero sottostringhe di una stringa lunga n con $m > 0$ simboli: $1 + n(n + 1)/2$

\mathbb{N} :

- linguaggi di tipo 0 ... 3 su Σ con $n > 0$ simboli
- funzioni calcolabili e funzioni che nel nome hanno “*ricorsive*”
- Numero di macchine di Turing

$\mathbb{P}(\mathbb{N})$:

- linguaggi su Σ con $n > 0$ simboli
- linguaggi non di tipo 0 ... 3 su Σ con $n > 0$ simboli
- funzioni parziali e totali

QUIZ SU CLASSI DI EQUIVALENZA

Vedere se rispetta le proprietà

- Tabella speculare: Binaria, riflessiva: (a, a) ; (b, b) e simmetrica: (a, b) ; (b, a) , ecc...
- Transitiva: $(a, b) \wedge (b, c) \Rightarrow (a, c)$

Per vedere le classi di equivalenza uguali raggruppare le lettere che stanno su righe uguali della tabella

QUIZ SU FUNZIONI RICORSIVE

- **Funzioni ricorsive di base:** costante 0, i-esima proiezione / identità, successore S
- **Funzioni primitive ricorsive:** sono anche totali
- **Funzioni calcolabili/ turing - calcolabili:** calcolate da una MdT in passi finiti

QUIZ SU RICONOSCIMENTO LINGUAGGI REGOLARI

Pattern corretti assieme \Rightarrow pattern totale corretto

Anche solo un pattern non corretto \Rightarrow pattern totale non corretto

riconoscimento:

- Regolari:
 - Linguaggio finito
 - Lunghezza delle stringhe cresce all'aumentare di n in modo costante
 - Puoi rappresentarlo attraverso un automa
- Non regolari:
 - Lunghezza cresce cresce all'aumentare di n in modo non lineare (geometrico o casuale)
 - Per poterlo riconoscere dobbiamo avere della memoria

QUIZ SU RICONOSCIMENTO IDENTITA' REGEXPR

Fondamentali

- $R + S = S + R$
- $(R + S) + T = R + (S + T)$
- $(RS)T = R(ST)$
- $(R + S)T = RT + ST$
- $\varepsilon = \emptyset^*$
- $\varepsilon = \varepsilon^*$
- $(\varepsilon + R)^* = R^*$
- $(\varepsilon + R^*R) = R^*$
- $R^{**} = R^*$

Avanzate

- $(R + S)^* = (R^*S^*)^* = (S^*R^*)^* = (R^*S^*)^*R^*$
- $(B^*A^*G^*N^*A^*R^*A^*) = (T^*E^*S^*T^*A^* + D^*I^* + C^*A^*Z^*Z^*O^*)^*$

Mai incontrate

- $R + \emptyset = R$
- $R\emptyset = \emptyset = \emptyset R$

QUIZ SU GRAMMATICHE

Grammatiche ambigue:

- $S \rightarrow SaS$ | a: due volte lo stesso simbolo in una regola di produzione
- $S \rightarrow aS$ | aSa | ϵ : una regola di produzione è = ad un'altra ma concatenata con simboli ulteriori
- $S \rightarrow Sa$ | aS | a: regola diversa ma che riporta allo stesso risultato
- $S \rightarrow Sa$ | a | ϵ : regola diversa ma che riporta allo stesso risultato x2

Forma normale di Chomsky: tutte le produzioni di una grammatica CF sono le seguenti:

- $A \rightarrow BC$ | a
- $S \rightarrow \epsilon$

Forma normale di Greibach: tutte le produzioni di una grammatica CF sono le seguenti:

- $A \rightarrow aA$

QUIZ SU LINGUAGGI

For: una funzione è for-calcolabile se essa è primitiva ricorsiva

Aliasing:

- Due variabili / espressioni hanno lo stesso l-value
- Due parametri sono passati per riferimento

QUIZ SU AUTOMI A PILA

Una scrittura del tipo: $\delta(q, a, B) = \{q, BA\}$ vuol dire che

- se mi arriva una a e sono nello stato q e in cima allo stack ho B
- Allora faccio una pop (B esce) e pusho BA (faccio una push di A e poi una push di B)
- Se prima dell'operazione avevo uno stack così: $B | A$ ora ho $B | A | A$

QUIZ SU MdT

Se lo stato della MdT è $\langle q_i, ua, b, cv \rangle$ sta a significare che :

- sono nello stato q_i ,
- la mia testina è su b (quindi ho letto b "poco fa"),
- A sx della mia testina ho una serie di caratteri uguali ad ua
- a dx invece ho una serie di caratteri uguali a cv

Successivamente, se ho una funzione di transizione del tipo: $\delta(q_i, b) = (q_j, c, R)$ vuol dire che

- se mi arriva una b e sono nello stato q_i
- Allora vado nello stato q_j sostituisco b con c e sposto la testina a dx (R = right)

A questo punto lo stato della mia MdT sarà: $\langle q_j, uac, c, v \rangle$