UTILIZZEREMO LINGUAGGI FORMALI → LINGUAGGI DI PROGRAMMAZIONE

LINGUAGGI ARTIFICIALI VENGONO CHIAMATI FORMALI E DOVREBBE ESSERE NON AMBIGUO, LA TRADUZIONE DEVE ESSERE AUTOMATICA

JAVASCRIPT AD ESEMPIO CONTINUA SOPRA GLI ERRORI "DON'T BREAK THE WEB"

→ DEVE GENERARE UN LOW CODICE COMPILATO

LA STRUTTURA IF, ELSE PUÒ ESSERE AMBIGUO SE NON UTILIZZO LE { }

QUANDO SI PRESENTA UN' AMBIGUITÀ IL COMPILATORE UTILIZZA DELLE POLICY PER RISOLVERLE, AD ES. ATTACCA L'ELSE ALL'ULTIMO IF.

FORMALE HA DELLE REGOLE CHE LO DEFINISCONO (COME NEI NATURALI)

Σ = INSIEME DEI SIMBOLI CHE COMPONGONO UNA FRASE (PAROLE)

= $\{a, b, c\}$ LINGUAGGIO COMPOSTO DA TRE SIMBOLI, $|\Sigma| = 3$

SONO AMMISSIBILI LINGUAGGI CON CARDINALITÀ ϕ → $L = \{\}$ INSIEME VUOTO

$L_3 = \{a^n b^n c^n, n \geq 1\}$ LE POSSIBILI PAROLE SONO ∞, $|L_3| = \infty$

IL NUMERO DI SIMBOLI CHE COMPONE UNA PAROLA = CARDINALITÀ PAROLA

$|ab| = 2$, ESISTE LA STRINGA VUOTA $|\varepsilon| = \phi$ UTILIZZO EPSILON

$L_4 = \{\varepsilon\}$ CARDINALITÀ $|L_4| = 1$

ALFABETO = COMPOSTO DA SIMBOLI MAI VUOTO E FINITO

LINGUAGGIO = COMPOSTO DA PAROLE DELL'ALFABETO PUÒ ESSERE VUOTO, FINITO O INFINITO

OPERATORI E OPERAZIONI

NEI LINGUAGGI FORMALI ESISTE LA "STRINGA VUOTA" $\Rightarrow |\varepsilon| = 0$ (EPSILON), PUÒ APPARTENERE O MENO AD UN LINGUAGGIO, ES.: $L_4 = \{\varepsilon, a, b\}$, $|L_4| = 3$

UN LINGUAGGIO CHE CONTIENE LA STRINGA VUOTA NON È UN LINGUAGGIO VUOTO MA HA CARDINALITÀ 1

STRINGHE

- **CONCATENAMENTO** → LA STRINGA VUOTA È "L'ELEMENTO NEUTRO"

$$\varepsilon \cdot x = x \quad , \quad x \cdot \varepsilon = x$$

- **POTENZA:**

$$x^n = \begin{cases} x^{n-1} x & n \geq 1 \\ x^0 = \varepsilon & n = 0 \end{cases}$$

$\boxed{\text{n SOLO POSITIVO}}$

$$(pa)^3 \neq pa^3$$

VALE SOLO SULL'ULTIMO ELEMENTO

- **RIFLESSIONE:** $(abc)^R = cba$, SE LO APPLICO AD UNA PALINDROMA NON VARIA $\Rightarrow (ama)^R = ama$

- **CLEAN:** GENERA INFINITE PAROLE CHE VANNO DA

$$x^* = x^0 \ldots x^n \ldots x^{+\infty}$$

ES.: $x = a$, $x^* = \varepsilon, a, aa, aaa, \ldots$

UTILIZZATO PER "COSE" POTENZIALMENTE INFINITE

LINGUAGGI

- **CONCATENAZIONE** $L_1 \times L_2 = \{x \cdot y \mid x \in L_1 \wedge y \in L_2\}$ OGNI PAROLA DI L1 VIENE CONCATENATA CON TUTTE LE PAROLE DI L2 → GENERO UN NUOVO LINGUAGGIO

$$L_1 = \{\varepsilon, a, b\} \quad , \quad L_2 = \{bb, aa\}$$

$$L_3 = L_1 \times L_2 = \{bb, aa, abb, aaa, bbb, baa\}$$

CARDINALITÀ DI L_3 $\quad |L_3| \leq |L_1| * |L_2|$

$L_1 = \{\varepsilon, a\}$, $|L_1| = 2$

$L_s = L_1 \times L_1 = \{\varepsilon, a, aa\}$

⊛ SE CONTIENE LA STRINGA VUOTA **NON** È LINGUAGGIO VUOTO !

CONCATENARE QUALCOSA CON UN LINGUAGGIO VUOTO GENERA **SEMPRE** UN LINGUAGGIO VUOTO $\Rightarrow L = \Phi \; , \; |L| = \phi$

- **ELEVAMENTO A POTENZA**

$$L^n = \begin{cases} L^{n-1} \cdot L \; , & n \geq 1 \\ L^0 = \{\varepsilon\} & n = 0 \end{cases}$$

- **CLEAN** ∘ **STAR** $\quad L^* = \bigcup_{n=\emptyset}^{\infty} L^n = L^0 \cup L^1 \cup L^2 \cup \dots L^{100} , \dots$

POSSO APPLICARLO ANCHE ALLE PAROLE E GENERA UN INSIEME INFINITO DI PAROLE

$$(a)^* = \varepsilon \cup a \cup aa \cup aaa \cup aaaa \cup \dots$$

$$L^* = (L^*)^* = (L^0 \cup L^1 \cup \dots L^{100} \cup \dots)^* = ((L^0)^* \cup (L^1)^* \cup (L^2)^* \cup \dots)$$

GENERO LINGUAGGI GIÀ GENERATI

- **OPERATORE +** $\Rightarrow$ INVECE CHE PARTIRE DA ZERO PARTE DA 1

APPLICATO AD UN LINGUAGGIO $\quad L^+ = L \cdot L^* \Rightarrow \bigcup_{n=1}^{\infty} L^n$

APPLICATO ALLE PAROLE $\quad (x)^+ = x(x)^*$

$$(a)^+ = a \cup aa \cup aaa \cup \dots$$

LA CARDINALITÀ NON CAMBIA POICHÉ È SEMPRE INFINITA $\infty - 1 = \infty$

- **NEGATO**: SONO TUTTE LE PAROLE CHE NON APPARTENGONO AD L_1 (DELLO STESSO ALFABETO) $\quad \overline{L_U} = \Phi$

LINGUAGGIO UNIVERSALE $\rightarrow L_U = \varepsilon^* - \{a, b\}^*$ $\qquad \overline{L_1} = L_U - L_1$

È A SUA VOLTA UN LINGUAGGIO

- **INTERSEZIONE**: $L_1 \cap L_2$
- **UNIONE**: $L_1 \cup L_2$
- **DIFFERENZA**: $L_1 - L_2$

SI UTILIZZANO I CONCETTI DEGLI INSIEMI MATEMATICI

I LINGUAGGI REGOLARI

POSSONO SEMPRE ESSERE DESCRITTI TRAMITE UN'ESPRESSIONE REGOLARE.

DEFINIZIONE PER CASI DI E.R. → ESPRESSIONI REGOLARI

- E.R. $= \emptyset$ INSIEME VUOTO
- E.R. $= \varepsilon$ STRINGA VUOTA
- E.R. $= a$ CON $a \in \Sigma$ QUALUNQUE SIMBOLO CHE $\in$ ALL'ALFABETO
- DEFINITE e_1, e_2 SONO E.R. ANCHE $e_1 \cup e_2$ È UNA E.R., ANCHE e_1^*, e_2^* SONO E.R., ANCHE $e_1 \bullet e_2$ È UNA E.R.

UN LINGUAGGIO REGOLARE SI BASA SU UNA ESPRESSIONE REGOLARE

ESEMPI DI ESPRESSIONI REGOLARI

- TUTTI I NUMERI INTERI POSITIVI O NEGATIVI CON E SENZA SEGNO (0 SENZA SEGNO)

$$\Sigma = \{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$ → ALFABETO SIGMA

$(+ \cup - \cup \varepsilon)(0 \cup 1 \cup 2 \cup \dots 9)^*$ ⇒ NO PERCHÉ POTREMMO AVERE NUMERI CHE INIZIANO CON 0 E NON VA BENE

$(+ \cup - \cup \varepsilon)(1 \cup \dots \cup 9)(0 \cup \dots \cup 9)^*$ ⇒ IN QUESTO CASO L'UNICO CASO CHE MANCA È LO 0 E VA AGGIUNTO

$$((+ \cup - \cup \varepsilon)(1 \cup \dots \cup 9)(0 \cup \dots \cup 9)^*) \cup (0)$$

- GENERATORE UN NUMERO CON O SENZA SEGNO IN FLOATING POINT CON ALMENO UNA CIFRA DI ESPONENTE

$$\Sigma = \{+, -, E, ., d\} \qquad d \rightarrow (0,1,2,3,4,5,6,7,8,9)$$

$$(+ \cup - \cup \varepsilon)(d) \cup (d \bullet \underbrace{dd^*}_{d^+}) \; E \; d \cdot d^*$$

+	3.845	E	18

SVILUPPO DEGLI OPERATORI / MECCANISMO DI DERIVAZIONE

- $(e_1 \cup \dots \cup e_n) \Rightarrow e_i$ CON $1 \leq i \leq n$
- $e^* \Rightarrow e \cdot e \cdot \dots \cdot e$ n VOLTE CON $0 \leq n$
- $e^+ \Rightarrow e \cdot e \cdot \dots \cdot e$ n VOLTE CON $1 \leq n$, e COMPARE ALMENO UNA VOLTA

ESEMPI

$$a^* (ab^* \cup c \cup aa)^* (a \cup b)^* c$$

UTILIZZO UNA POLITICA LEFT-MOST

$$\Rightarrow aaa (ab^* \cup c \cup aa)^* (a \cup b)^* c$$

$$\Rightarrow aaa \boxed{(ab^* \cup c \cup aa)} (ab^* \cup c \cup aa) (a \cup b)^* c$$

SCELGO c

$$\Rightarrow aaac \boxed{(ab^* \cup c \cup aa)} (a \cup b)^* c$$

SCELGO ab^*

$$\Rightarrow aaacab^* (a \cup b)^* c$$ SCELGO LA STRINGA VUOTA

$$\Rightarrow aaaca (a \cup b)^* c$$ SCELGO $\emptyset$ PER LA STAR

$$\Rightarrow aaacac$$

LINGUAGGIO CON PALINDROME

$\Sigma = \{a, b\}$ UTILIZZIAMO PALINDROME SENZA MARCA DI CENTRO OVVERO CON VALORI PARI, DEVONO RIFLETTERSI ab~~ba~~

UNA GRAMMATICA È UNA SERIE DI REGOLE CHE MI PERMETTE DI GENERARE TUTTO QUELLO CHE APPARTIENE AD UN LINGUAGGIO

PAL $\rightarrow$ a PAL a o PAL $\rightarrow$ b PAL b
PAL $\rightarrow$ aa e PAL $\rightarrow$ bb

QUESTA È UNA GRAMMATICA LIBERA DAL CONTESTO, OVVERO C'È SOLO UN NON TERMINALE A SINISTRA DELLA FRECCIA MENTRE A DX C'È UNA COMBINAZIONE VARIEGATA SENZA VINCOLI DI TERMINALI E NON TERMINALI.
I LINGUAGGI LIBERI DAL CONTESTO NON PIÙ ESPRESSIVI DELLE ESPRESSIONI REGOLARI, O DA QUELLI CHE DIPENDONO DAL CONTESTO

PARTENDO DA UNA GRAMMATICA $G = < V, \Sigma, P, S >$ DEFINISCO UN LINGUAGGIO $L(G) = \{ x \in \Sigma^* \mid S \xrightarrow{*} x \}$, PER UN LINGUAGGIO HO INFINITE GRAMMATICHE CHE LO POSSONO GENERARE

ES.: $V = \{S, A, B\}$, $\Sigma = \{a, b\}$, $S \to aA \mid S$

$A \to S \qquad S \to A \to S \qquad B \to a$

GRAMMATICA RIDOTTA

1- $\forall A \in V \quad \neg\exists \; A \xrightarrow{+} A$ PER QUALUNQUE NON TERMINALE NON ESISTE IL FATTO CHE RITORNI AL NON TERMINALE STESSO

2- $\forall A \in V \quad L(A) = \not\emptyset$ SE PER QUALUNQUE NON TERMINALE, SE LO PRENDO COME ASSIOMA NON GENERO MAI IL LINGUAGGIO VUOTO

3- $\forall A \in V$ QUALUNQUE NON TERMINALE È RAGGIUNGIBILE DALL'ASSIOMA

$$S \xrightarrow{+} \alpha A \beta \quad \text{con } \alpha, \beta \in (V \cup \Sigma)^*$$

ESEMPIO

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid n$

SE UNA GRAMMATICA È AMBIGUA NON È DETTO CHE ANCHE IL LINGUAGGIO SIA AMBIGUO; AMBIGUO $\Rightarrow$ DATA UNA PAROLA ESISTONO DUE O PIÙ ALBERI DI DERIVAZIONE

$$L = \{ a^* b^n c^n \cup a^n b^n c^* , n > 0 \}$$

GRAMMATICHE LINEARI

NELLA PARTE DESTRA SE C'È UN NON TERMINALE NON CE NE POSSONO ESSERE ALTRI MENTRE CI POSSONO ESSERE TERMINALI PRIMA O DOPO

$$V \times (\Sigma^* V \Sigma^*) \cup (\varepsilon)$$

SI DIVIDONO IN GRAMMATICHE LINEARI DX E SX, È UN RESTRINGIMENTO DELLA GRAMMATICA INDICA IL POSIZIONAMENTO DEL NON TERMINALE

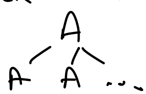
SINISTRA: $V \times \Sigma^* \cup \varepsilon$

DESTRA: $V \times \Sigma^* V \cup \varepsilon$

DX E SX SONO EQUIVALENTI SI POSSONO TRASFOR. UNA NELL'ALTRA

SONO PREFERIBILI LE DESTRE PER UN PROBLEMA DI RICORSIONE DELLE SX

$A \to A \ldots.$

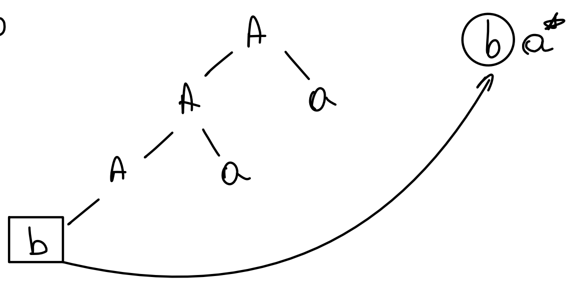


→ RICORSIONE INFINITA

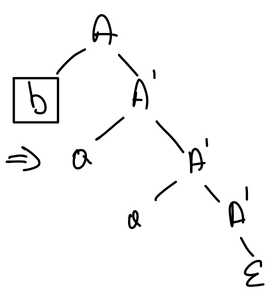
METODO PER ELIMINARE RICORSIONE SX

VALE ANCHE PER LE GRAMMATICHE NON LINEARI, (RICORSIONE IMMEDIATA)

$A \to Aa \mid b$



VOGLIO ARRIVARE AD AVERE UN ALBERO DELLA RICORSIONE ⇒



$$\left[\begin{array}{l} A \to b \\ A \to bA' \end{array}\right.$$

$A' \to aA'$

$A' \to \varepsilon$

$A \rightarrow A\beta_1 \mid \dots \mid A\beta_n$

$A \rightarrow \gamma_1 \mid \dots \mid \gamma_m$

$\beta_1 \dots \beta_n \in (\Sigma \cup V)^*$

$\gamma_1 \dots \gamma_m \in (\Sigma \cup V)^*$ MA NON INIZIANO CON A

PRIMO METODO

$A \rightarrow \gamma_1 A' \mid \dots \mid \gamma_m A'$

$A' \rightarrow \beta_1 A' \dots \mid \beta_n A' \mid \varepsilon$

SECONDO METODO

$A \rightarrow \gamma_1 A' \mid \dots \mid \gamma_m A'$

$A \rightarrow \gamma_1 \mid \dots \mid \gamma_m$

$A' \rightarrow \beta_1 A' \mid \dots \mid \beta_n A' \mid \beta_1 \mid \dots \mid \beta_n$

ENTRAMBI I METODI CREANO UNA RICORSIONE A DX

ESEMPIO

$R \rightarrow Ra \mid RTa \mid ab \mid c$

$R \rightarrow abR' \mid cR'$ TUTTE LE REGOLE DI R

$R' \rightarrow aR' \mid TaR' \mid \varepsilon$

PRIMO METODO

$R \rightarrow ab \mid c \mid abR' \mid cR'$

$R' \rightarrow a \mid Ta \mid aR' \mid TaR'$

SECONDO METODO

ESERCIZIO

$A \rightarrow Bx \mid A\boxed{yc} \mid x \mid y \mid A\boxed{c}$

$A \rightarrow BxA' \mid xA' \mid yA'$

$A' \rightarrow \varepsilon \mid ycA' \mid cA'$

AVRÒ TRE REGOLE PER A, ANTICIPO LA RICORSIONE E FACCIO PARTIRE LA RICORSIONE (HO APPLICATO IL SECONDO METODO)

$A \rightarrow Bx \mid x \mid y \mid BxA' \mid xA' \mid yA'$

$A' \rightarrow ycA' \mid cA' \mid yc \mid c$

(APPLICO IL PRIMO METODO)

HO TRE REGOLE, TRE CHIUSURE DELLA RICORSIONE

NON SEMPRE SONO OBBLIGATO AD UTILIZZARE LA STESSA REGOLA AD OGNI PASSO DELLA RICORSIONE

ESERCIZIO

UTILIZZO IL PRIMO METODO

$C \rightarrow \boxed{Ca} \mid \boxed{CCa} \mid a \mid aC$

$C \rightarrow a \mid aC \mid aCC' \mid aC'$

$C' \rightarrow Ca \mid a \mid CaC' \mid aC'$

NELLE REGOLE IN CUI COMPARE IL C' LO VADO A METTERE TUTTO A DESTRA

$C \rightarrow aC' \mid aCC'$

UTILIZZO IL SECONDO METODO

$C \rightarrow \varepsilon \mid CaC' \mid aC'$

$L(\text{CONTEXT FREE}) > L(\text{LINEARI}) > L(\text{LINEARI DX / SX})$

$= L(\text{STRETTAMENTE LINEARI DX, SX}) = L(\text{REGOLARI})$

$< L(\text{ESPRESSIONI REGOLARI})$

LINGUAGGIO LINEARE

HANNO UNA CARATTERISTICA SPECIFICA: NELLA PARTE DX DELLA REGOLA DI PRODUZIONE C'È SE APPARE AL PIÙ UN NON TERMINALE

$$V \times \left(Z^* \left(V \cup \{\varepsilon\} \right) Z^* \right)$$

$\forall\ L(\text{LINEARE})$ È UN $L(\text{CONTEXT FREE})$

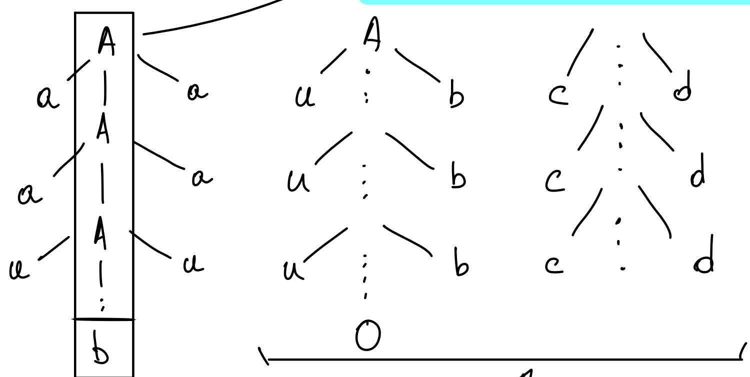
NELLA LINEARE ESISTE UN SOLO ASSE DI CRESCITA

$A \to aAa \mid b$

$a^n b a^n$

$a^n b^n < b^n a^n$

$\boxed{a^n b^n \mid c^k a^k}$

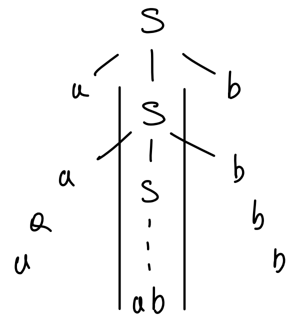


IN UN L(CONTEXT FREE)
NON HO PROBLEMI:

$S \to A\ B$ → REGOLA NON LINEARE

$A \to aAb \mid \varepsilon$
$B \to cBd \mid \varepsilon$ → QUESTE DUE REGOLE SONO LINEARI

$S \to aSb \mid ab$

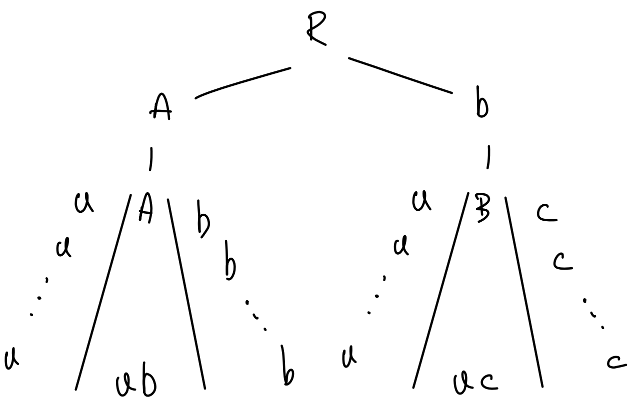


$R \to AB$

$A \to aAb \mid ab$

$B \to aBc \mid ac$

ALBERO DI DERIVAZIONE



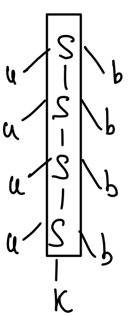
ESISTONO DELLE GRAMMATICHE CONTEXT FREE CHE SONO NON LINEARI

$L(\text{LINEARE}) > L(\text{LINEARE DX}, \text{SX})$

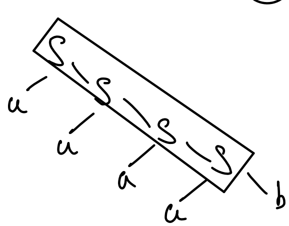
AGGIUNGE UN VINCOLO SUL POSIZIONAMENTO DEL NON TERMINALE

DX: $V \times (\Sigma^*(V \cup \{\varepsilon\}))$

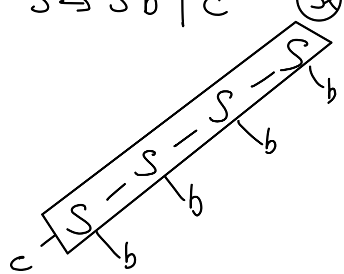
$S \to aSb \mid K$ GRAMMATICA LINEARE



$S \to aS \mid b$ DX



$S \to Sb \mid c$ SX



$S \to aS \mid kB$
$B \to bB \mid \varepsilon$
→ GRAMMATICA LINEARE DX PER $\boxed{a^n k b^n}$ ≠

$S \Rightarrow aaaaakB \Rightarrow aaaaakbB \Rightarrow aaaaakb$ PAROLA LECITA x LA GRAMMATICA

L (LINEARE DX E SX) = L (STRETTAMENTE LINEARE DX, SX)

VINCOLO SUI TERMINALI: SE APPAIONO DEI TERMINALI AL PIÙ NE APPARE UNO

ES.: GRAMMATICA STRETTAMENTE LINEARE DX

$S \to a$
$S \to b$
$S \to B$
$S \to aB$
$S \to \varepsilon$

$S \to abc$
$S \to abcB$
$S \to a_1 \dots a_n$
$S \to a_1 \dots a_n B$

REGOLE LINEARI MA NON STRETTAMENTE LINEARI

$S \to aS'$ $\quad S \to bS''$
$S' \to c$
→ HO TRASFORMATO IN STRETTAMENTE LINEARE

$S \to a_1 S'$

$S'' \to a_2 S'''$

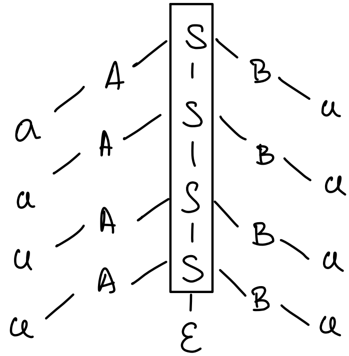
$S \to aSb$ REGOLA RICORSIVA AUTO INCLUSIVA

$S \to aS$ REGOLA RICORSIVA NON-AUTO INCLUSIVA

SE LA GRAMMATICA HA REGOLE RICORSIVE AUTOINCLUSIVE IL LINGUAGGIO POTREBBE ESSERE NON REGOLARE

$S \to ASB \mid \varepsilon$
$A \to a$
$B \to a$
→ LINGUAGGIO REGOLARE MONOTERMINALE, IN UN LINGUAGGIO CONTEXT-FREE I MONOTERMINALI SARANNO SEMPRE REGOLARI

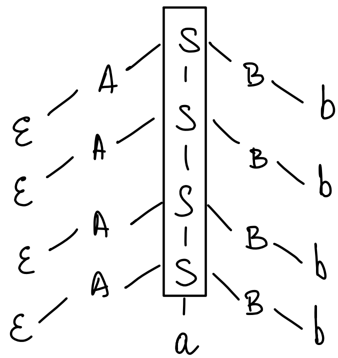
$$S \to ASB \mid \varepsilon$$
$$A \to a$$
$$B \to a$$



==LINGUAGGIO MONOTERMINALE== E QUINDI REGOLARE

$$a^n a^n = a^{2n}$$

$$S \to A S B \mid a$$
$$A \to \varepsilon$$
$$B \to b$$



REGOLA RICORSIVA AUTOINCLUSIVA

$$S \to AB\boxed{S}CD$$

$$S \to aaSbbb \mid aSbbb \mid aaaSbbb$$
$$S \to \varepsilon$$

ESSENDOCI UN VINCOLO NON È REGOLARE PERCHÉ C'È UN BILANCIAMENTO

AD. ES. DATE bbb POSSO GENERARE DA 1 A 3 a

$$\Downarrow$$

$$a^n b^{3m}, \quad m \leq n \leq 3m$$

$$a^n \left(a \cup \left(b \cup ad \cup a^n \right) b^n \left(c \cup a^* \right) \ldots \right)$$

HO TROVATO UN BILANCIAMENTO QUINDI ESSENDOCI UN VINCOLO IL LINGUAGGIO NON È REGOLARE (ANCHE SOLO DUE ELEMENTI), HO UNA QUALUNQUE ESPRES. MATEMATICA CHE MI COLLEGA DUE O + ESPONENTI

$S \rightarrow ASB \mid \varepsilon$

NON È REGOLARE !

$A^n \quad B^n$

$(a^+)^n \underline{(c \cup b)}^n$ → PARTO A GUARDARE DA DX

$A \rightarrow aA \mid a \Rightarrow A^+$ (O UNA a O UN NUMERO TOT. DI a)
ALMENO UNA A

$B \rightarrow c \mid b \Rightarrow (c \cup b)$

$\Downarrow$ MODIFICA ALLA REGOLA PER CAMBIO ESEMPIO

$A \rightarrow aA \mid \varepsilon \Rightarrow A^*$ $\quad (A^*)^n (c \cup b)^n$

LA MODIFICA È REGOLARE PERCHÉ POSSO SAPERE IL NUMERO DI C E DI B MA NON SO NULLA SU QUANTE A^*, LA PARENTESI DX NON VINCOLA LA SX DIVERSAMENTE DA PRIMA

$$\boxed{\begin{array}{l} S \rightarrow aS \mid bS \mid A \\ A \rightarrow Ab \mid c \end{array}}$$

LA GRAMMATICA CONTIENE UNA REGOLA RICORSIVA AUTOINCLUSIVA, MA QUESTO CASO SU A (NON CAMBIA NULLA)

$$\boxed{S \rightarrow aS \mid bS \mid c}$$

NON HA REGOLE RICORSIVE AUTOINCLUSIVE, IL LINGUAGGIO È REGOLARE, GRAMMATICA REGOLARE

$$\boxed{\begin{array}{l} S \rightarrow Ab \mid AS \\ A \rightarrow a \mid Aa \mid bA \end{array}}$$

NON HA REGOLE RICORSIVE AUTOINCLUSIVE, IL LINGUAGGIO È REGOLARE, GRAMMATICA REGOLARE

VERIFICO SE:

1) MONOTERMINALE
2) REGOLE RICORSIVE AUTOINCLUSIVE, $\boxed{A \rightarrow \alpha A \beta}$

L(LINEARI SX / DX) ⊆ L(REGOLARI)

* SOLO PARTE DELLA DIMOSTRAZ. PER MANCANZA STRUMENTI

CONSIDERO I LINEARI DX

$$X \to \alpha_1 \mid \dots \mid \alpha_n \Rightarrow \boxed{X = \alpha_1 \cup \dots \cup \alpha_n}$$

LA VEDO COME UN'ESPRESSIONE 1°

$$X = \boxed{\gamma} X \cup \boxed{\beta} \Rightarrow X = \gamma^* \beta$$

ESEMPIO:

$$X \to \underbrace{\boxed{a}}_{\gamma} X \mid \underbrace{\boxed{b}}_{\beta} \Rightarrow X = a^* b$$

HO TRASFORMATO LA REGOLA DI PRODUZIONE OTTENENDO UN'ESPRESSIONE REGOLARE

AVENDO PIÙ NON TERMINALI MI RITROVERÒ UN SISTEMA DI EQ. CHE ANDRÒ A RISOLVERE CON IL METODO DELLA SOSTITUZIONE

$$\begin{cases} S \to aB \mid aS \\ B \to dS \mid b \end{cases} \Rightarrow \begin{cases} S = \underbrace{\boxed{aB}}_{\beta} \cup \underbrace{\boxed{a}}_{\gamma} S \\ B = dS \cup b \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} S = a^* a B \\ B = dS \cup b \end{cases} \Rightarrow \begin{cases} S = a^* a (dS \cup b) \\ B = dS \cup b \end{cases} \Rightarrow$$

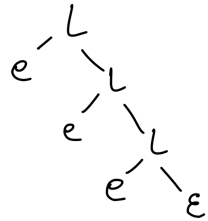
$$\Rightarrow \begin{cases} S = \underbrace{\boxed{a^* a d}}_{\gamma} S \cup \underbrace{\boxed{a^* a b}}_{\beta} \\ B = dS \cup b \end{cases} \Rightarrow \begin{cases} S = (a^* a d)^* a^* a b \\ B = dS \cup b \end{cases}$$

ESERCIZI SCRIVIAMO DELLE GRAMMATICHE

e^*

$L \to e\,L$

$L \to \varepsilon$



SCRIVERE UN NUMERO BINARIO DI LUNGHEZZA ARBITRARIA, ANCHE VUOTO

$L \to \varepsilon$

$L \to \emptyset L \mid 1 L$

→ VA BENE MA SI PUÒ MIGLIORARE PER SUPPORTARE VARIAZIONI NELLE RICHIESTE

$L \to \varepsilon$ $\quad$ $L \to N L$ $\quad$ UTILIZZO UN NON TERMINALE

$N \to \emptyset \mid 1$

SE VOGLIAMO TRADURRE UN LINGUAGGIO DI PROGRAM. DEVO CONSIDERARE ANCHE I SEPARATORI ES.: ;

$L \to N; \mid N;\,L$ $\quad$ AD. ES. IN C TERMINA TUTTO CON IL ;

SE AVESSI PIÙ SEPARATORI AGGIUNGO UN NON TERMINALE CHE MI DESCRIVE:

$L \to NX \mid NXL$, $X \to ; \mid , \mid :$

IPOTIZZO DI NON METTERE UN SEPARATORE ALLA FINE, MA POSSO AVERE LISTE VUOTE

$L \to \varepsilon \mid N X L$, $X \to ;$, $N \to a$ (PRIMA E DOPO LA X C'È UN'ISTRUZIONE)

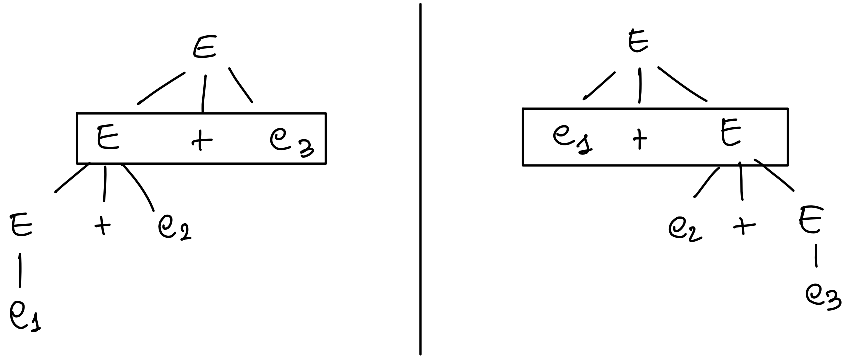
$L \to a;$ $\quad L \to a;a;$ $\quad L \to a;a;a;$ $\quad L \to a;a;a\boxed{;}$ NON VA BENE NON DEVE TERMINARE ;

NEL CASO DI LISTA NON VUOTA: $L \to N \mid NXL$, MA PER LA LISTA VUOTA AGGIUNGO UNA REGOLA: $L' \to \varepsilon \mid L$

LISTA DI ISTRUZIONI COMPRESE TRA UN SIMBOLO DI INIZIO E FINE

$S \to S_{START}\ L\ S_{END}$

$L \to e;L \mid \varepsilon$



GRAZIE ALLA PROPR. ASSOCIATIVA IL RISULTATO È UGUALE
⇒ ASSOCIATIVITÀ DESTRA O SINISTRA

$E \to E+E \mid E*E \mid e$ È UNA GRAMMATICA AMBIGUA PERCHÉ SE CERCO DI COSTRUIRE ALBERI DI DERIVAZIONE POSSO GENERARNE DUE
MA IN QUESTO CASO SONO DUE INTERPRET. DIVERSE, RISULTATI DIVERSI,

$E \to E+E \mid E*E \mid (E) \mid e$ È ANCORA UNA GRAMMATICA AMBIGUA

$E \to (E+E) \mid (E*E) \mid (E) \mid e$ AGGIUNGENDO A TUTTE LE OPER. LE TONDE OBBLIGO A SCRIVERE ESPLICITAMENTE LE PRIORITÀ.

SE HO OPERATORI CON DIVERSA PRIORITÀ NON POSSO AVERE UNA GRAMMATICA SU UN'UNICO LIVELLO, MA DI UNA SUDDIVISA IN PIÙ LIVELLI

$\boxed{E} \rightarrow E \oplus T \mid T$

$T \rightarrow T \circledast F \mid F$

$F \rightarrow e \mid (E)$

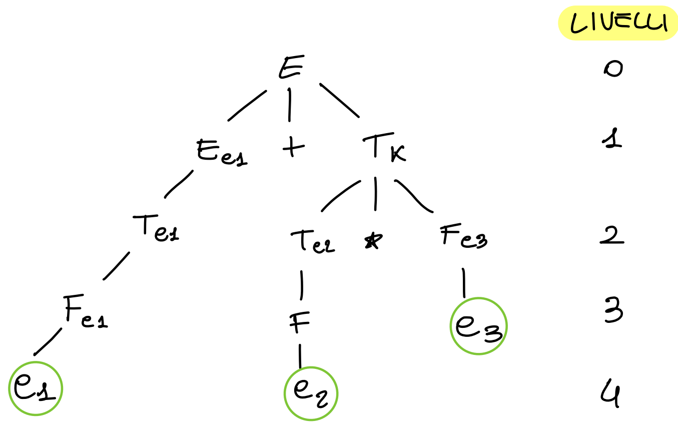
E È IL MIO ASSIOMA, IL +, ×, ( ) DEVONO STARE SU TRE LIVELLI DIVERSI

AD OGNI LIVELLO CORRISPONDE UNA PRIORITÀ
CHI HA PRIORITÀ MINORE STA PIÙ IN ALTO

$E \rightarrow E - T$ AGGIUNGO IL MENO ALLO STESSO LIVELLO DEL +

$T \rightarrow T / F$ AGGIUNGO LA / ALLO STESSO LIVELLO DEL *

$K = T_{e_2} * F_{e_3}$



$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow e \mid (E)$

NELLA GRAMMATICA ALGEBRICA SCENDO DI UN LIVELLO PER VOLTA

COSTRUIRE L'ALBERO DI DERIVAZIONE DI: $e_1 / e_2 * e_3$
(ASSOCIATIVA A SINISTRA)

AGGIUNGERE UN OPERATORE @ ASSOCIATIVO A SX, CON PRIORITÀ MAGGIORE DI + E - MA MINORE DI * E /.

DEVO ANDARE AD AGGIUNGERE UN LIVELLO

$E \rightarrow E + K \mid E - K \mid K$

$K \rightarrow K @ T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow e \mid (E)$

AGGIUNGERE UN OPERATORE BINARIO # ASSOCIATIVO A DX CON PRIORITA' INFERIORE RISPETTO A + E -

$\boxed{P \rightarrow E \# P \mid E}$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow e \mid (P)$

DEVO CAMBIARE SEMPRE L'ASSIOMA PERCHÉ HO INTRODOTTO UN LIVELLO ANTECEDENTE AL PRIMO

AGGIUNGO UN LIVELLO DOPO L'ULTIMO AD. ES. INTRODUCO LE [ ]

$P \rightarrow E \# P \mid E \mid$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (G) \mid G$

$G \rightarrow e \mid [P]$

GRAMMATICA PER LA NOTAZIONE POLACCA INVERSA

$E \rightarrow EE+ \mid EE* \mid EE- \mid EE/$

$E \rightarrow e$

GIA' LA NOTAZIONE MI INDICA LA PRIORITA' DEGLI OPERATORI

ESERCIZI: DATA UN'ESPRESSIONE SCRIVERE LA CORRISPONDENTE GRAMMATICA CONTEXT-FREE

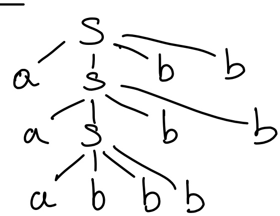
$a^n c b^n \quad n \geq 2$

C'È UN BILANCIAMENTO $\Rightarrow$ DEVO SCRIVERE UNA REGOLA RIC. AUTOINC.

$S \rightarrow aSb \mid aacbb$

$a^n b^{2n+1} \quad n \geq 1 \quad = a^n b b^{2n}$

$S \rightarrow aSbb \mid abbb \qquad b^{2n+1} = b^{2n} \cdot b$



MI È COMODO VEDERE LA B AL CENTRO $= b \cdot b^{2n}$

$a^n b^{2n} b \quad n \geq 1$

$S \rightarrow Ab$

$A \rightarrow aAbb \mid abb$

MODI DIVERSI PER INTERPRETARE LA STESSA ESPRESSIONE GRAMMATICHE DIVERSE MA CORRETTE

$\underbrace{a^n \underbrace{b^m c^m}_{Y} d^n}_{X} \quad n, m > 1$

- DUE BILANCIAMENTI UNO DENTRO L'ALTRO, DEVO AVERE DUE NON TERMINALI
- GENERO PRIMA QUELLO ESTERNO

$X \rightarrow aXd \mid aaYdd$

$Y \rightarrow bYc \mid bbcc$

$\underbrace{a^n b^n}_{X} \underbrace{c^m d^m}_{Y} \quad n, m \geq 0$ • UTILIZZO DUE NON TERMINALI

$S \to X Y$ ⇒ REGOLA CHE LEGA I DUE NON TERMINALI

$X \to a X b \mid \varepsilon$

$Y \to c Y d \mid \varepsilon$

$\underbrace{a^n \boxed{c^*} d^n}_{} \quad n \geq 0$ • UTILIZZO COMUNQUE DUE NON TERMINALI PER GESTIRE SIA c^* SIA IL BILANCIAMENTO

$S \to a S d \mid C$

$C \to c C \mid \varepsilon$ → È UNA LISTA CHE AMMETTE ZERO ELEMENTI

LE TANTE PARTI CHE COMPONGONO IL CASE SONO DELLE LISTE CON SEPARATORE (NON VUOTA!)

CASE id OF

$\boxed{\text{num , num , num , S}}$; → E ELEMENTO DELLA LISTA — CASO 1

num , num : CASE id OF — CASO 2
num, num , S;
num , S
END CASE;

num : S — CASO 3

→ L

END CASE

STATEMENT → S | CASE

CASE → CASE id OF L END CASE → NON TERMINALE CHE GESTISCE IL CORPO DEL PRIMO CASE (CASO 1,2,3)

L → E ; L | E → HO DEFINITO LA STRUTTURA DELLA LISTA NON I CASE

E → LNUM : STATEMENT → NON TERMINALE CHE MI DEFINISCE LA LISTA DEI num

LNUM → num , LNUM | num → LISTA NON VUOTA DI num

```
CASE id OF

   num : CASE id OF
            num : S ;
            num , num : S
         END CASE ;
   ------------------------------------------
   num , num : CASE id OF
                 num : S ;
                 num , num : S
               END CASE
   ------------------------------------------
END CASE
```

CASO 1

CASO 2 → AVRÒ UNA LISTA L DI DUE ELEMENTI

SCRIVERE UNA GRAMMATICA CONTEXT FREE CHE DESCRIVA:

$$\underbrace{(a \cup b)^n \, c^* \, d^{2n+1}}_{A} \; \underbrace{a^J \underbrace{b c^K}_{C} d^m}_{B} \quad ⊛ \qquad n > \emptyset, \; J = K + m, \; m, K \geq 0$$

- TROVO I BILANCIAMENTI: HO 3 BILANCIAMENTI, NON POTRÒ MAI AVERE IN UNA GRAMMATICA CONTEXT FREE UN BILANCIAMENTO
- REGOLA ASSIOMATICA: $S \to A\,B$

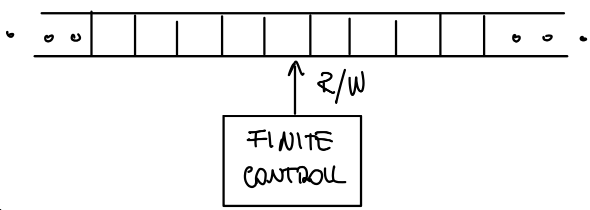
$$
\boxed{
\begin{array}{l}
A \to T\,A\,dd \mid T\,K\,ddd \\
T \to a \mid b \\
K \to c\,K \mid \varepsilon \\
B \to a\,B\,d \mid C \\
C \to a\,C\,c \mid b
\end{array}
}
$$

⊛ GESTISCO SEMPRE DALL'ESTERNO ALL'INTERNO (ANCHE CON UN SOLO BILANCIAM.)

GERARCHIA DI NOAM CHOMPSKY

CLASSE GRAMMATICA	REGOLE	CLASSE LINGUAGGIO	CLASSE METODO RICORSIONE
TIPO 3	$A \to aX$ $a \in \Sigma^*$ LINEARI DX/SX	LINGUAGGI REGOLARI	AUTOMI A STATI FINITI
TIPO 2	$A \to a$ $a \in (Z \cup V)^*$	CONTEXT FREE	AUTOMI NON DETERMINISTICI A PILA
TIPO 1	$\alpha A \beta \to \alpha \gamma \beta$ $\gamma \neq \varepsilon$	CONTEXT SENSITIVE	MACCHINA DI TURIN (TEMPO FINITO)
TIPO $\emptyset$	$\alpha A \beta \to \alpha \gamma \beta$	RICORSIVAMENTE ENUMERATIVA	MACCHINA DI TURIN

MACCHINA DI TURIN



HA TRE ELEMENTI:

- UN NASTRO INFINITO USATO PER INPUT E MEMORIA DI SUPPORTO
- UNA TESTINA, LEGGE | MODIFICA IL CONTENUTO DI UNA CELLA
- UN POSTO IN CUI RISIEDE IL PROGRAMMA (FINITE CONTROL)

DAGLI AUTOMI A PILA SI INTRODUCE LA MEMORIA AUSILIARIA

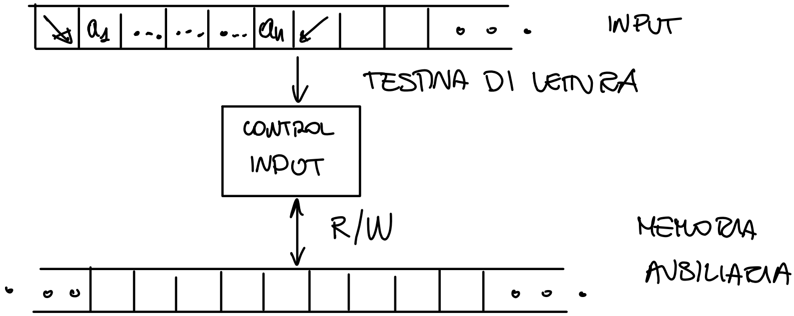


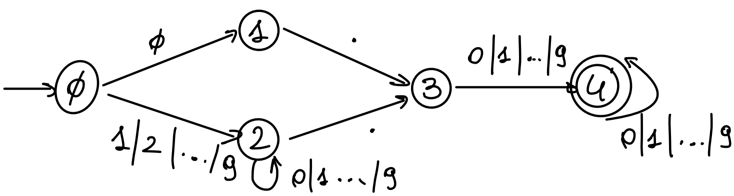
TABELLA PER LA CONTROL INPUT

STATI

	S_0	S_1	...	S_K
q_0				
q_1				
⋮				
q_n				

- IN OGNI CELLA CI POSSONO ESSERE UNA O PIÙ ISTRUZIONI
- SE CI SONO AL PIÙ UNA ISTRUZIONE PER CELLA L'AUTOMA È DETERMINISTICO
- SE ALMENO UNA CELLA NE CONTIENE DUE O PIÙ È NON DETERMINISTICO

NOTAZIONE GRAFICA PER GLI AUTOMI A STATI FINITI



UTILIZZO UN GRAFO

SE TERMINO SULLO STATO FINALE (TERMINO CON UN SI, NO ALTRIMENTI) È UNO STATO ACCETTANTE, POSSO AVERE PIÙ STATI FINALI (ALMENO UNO), HO UNO SOLO STATO INIZIALE CHE PUÒ COINCIDERE CON UNO FINALE.

POSSO AVERE DELLE LABEL CON DEI SIMBOLI CHE MI PERMETTONO DI PASSARE DA UN NODO ALL'ALTRO (ES.: ∅)

L'AUTOMA IN ESEMPIO VA A RICONOSCERE UN QUALSIASI NUMERO CON VIRGOLA

QUANTO COSTA IN TERMINI DI TEMPO RICONOSCERE UNA PAROLA?

IN UN AUTOMA A STATI FINITI LA COMPLESSITA' È LINEARE NELLA LUNGHEZZA DELLA PAROLA (FINCHÉ NON RAGGIUNGO UNO STATO FINALE E HO CONSUMATO TUTTI GLI INPUT)

DEFINIZIONE FORMALE DI AUTOMA

È DEFINITO DA 5 ELEMENTI: $\langle Q, \Sigma, \delta, q_\emptyset, F \rangle$

- Q È L'INSIEME DEGLI STATI
- $q_\emptyset$ È LO STATO INIZIALE
- F SOTTOINSIEME DI Q, L'INSIEME DEGLI STATI FINALI
- Σ È L'ALFABETO
- δ FUNZIONE DI TRANSIZIONE, QUANTI ARCHI CI SONO E CHE LABEL HANNO

CAMBIANDO LA FUNZIONE CON DEI VINCOLI O SU COME È SCRITTA CAMBIO IL TIPO DI AUTOMA E QUINDI QUELLO CHE PUÒ RICONOSCERE

$\langle Q, \Sigma, \delta, q_\emptyset, F \rangle$

$\delta : Q \times \Sigma \to Q$ VINCOLO PER AUTOMA A STATI FINITI DETERMINISTICO

	S_0	S_1	S_2	...	S_n	
$q_\emptyset$	q_{12}	/				
q_1						
q_2	q_3					
⋮						
q_n						

PARTENDO DA UNA GRAMMATICA STRETTAMENTE LINEARE GENERARE L'AUTOMA A STATI FINITI RICONOSCITORE

LA GRAMMATICA È DESCRITTA DA $\langle V, \Sigma, P, S \rangle$

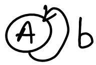
L'AUTOMA È DESCRITTO DA $\langle Q, \Sigma, \delta, q_0, F \rangle$

- PER OGNI NON TERMINALE HO UNO STATO, LO STATO INIZIALE q_ϕ COINCIDE CON L'ASSIOMA: $q_\phi \equiv S$

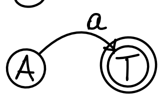
ES.:

$A \to aB$

$A \to bA$

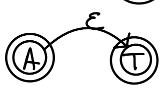


$A \to a$

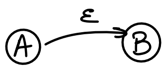


PER GESTIRE QUESTO TIPO DI REGOLE AGGIUNGO UN NON TERMINALE CHE SARÀ UNO STATO FINALE

$A \to \varepsilon$



$A \to B$



POSSO MUOVERMI DA A VERSO B SENZA CONSUMARE INPUT

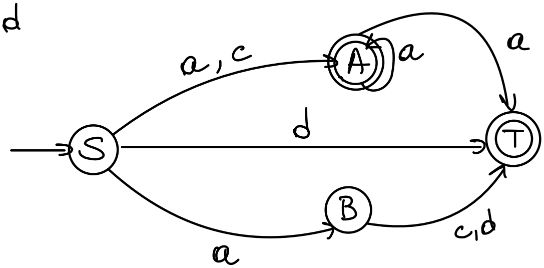
ESERCIZIO

$S \to aA \mid aB \mid cA \mid d$

$A \to aA \mid a \mid \varepsilon$

$B \to c \mid d$

- CREO TANTI STATI QUANTI I NON TERMINALI
- GESTISCO TUTTE LE REGOLE DI PRODUZIONE SINGOLARMENTE



IN QUESTO CASO L'AUTOMA È NON DETERMINIS.

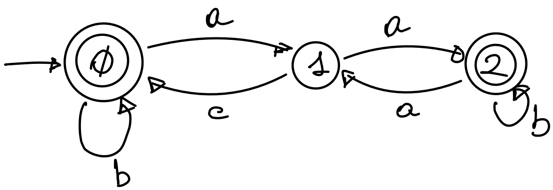
AUTOMI A STATI FINITI

DEF. DETERMINISTICA:

$\langle Q, \Sigma, \delta, q_\emptyset, F \rangle$

$\delta: Q \times \Sigma \rightarrow Q$

IL RICONOSCIMENTO AVVIENE SE DOPO AVER CONSUMATO TUTTI GLI INPUT MI TROVO IN UNO STATO FINALE ALTRIMENTI NON RICONOSCO



baab APPARTIENE AL LINGUAGGIO?

INPUT	STATO
$\boxed{b}aab$	$\emptyset$
$\boxed{a}ab$	$\emptyset$
$\boxed{a}b$	1
$\boxed{b}$	2
/	2

→ CONSUMO L'INPUT b E VADO IN $\emptyset$

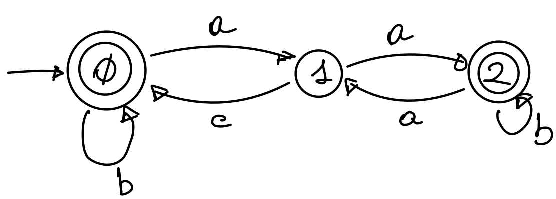
→ CONSUMO a E VADO IN 1

→ CONSUMO a E VADO IN 2

→ CONSUMO b E VADO IN 2

→ NON HO PIÙ SIMBOLI E SONO IN UNO STATO FINALE LA PAROLA $\in$ AL LINGUAGGIO

INPUT	STATO
$\boxed{a}aab$	$\emptyset$
$\boxed{a}ab$	1
$\boxed{a}b$	2
b	1



→ NELLO STATO 1 NON C'È UN ARCO USCENTE CON LABEL b, IL LINGUAGGIO / AUTOMA NON PREVEDE b IN QUESTO PUNTO → FALLIMENTO

* POSSO GESTIRE CON QUESTI AUTOMI PAROLE IN INPUT DI QUALSIASI DIMENSIONE, APPLICO LA FUNZIONE DI TRANSIZIONE TANTE VOLTE QUANTI SONO I SIMBOLI IN INPUT ⇒ COMPLESSITÀ LINEARE

$\delta^*: Q \times \Sigma^* \longrightarrow Q \qquad q \in Q$

$\delta^*(q, ya)$ APPLICO LA FUNZIONE DI TRANSIZIONE δ^* A PARTIRE DA q CON LA STRINGA ya $\Rightarrow a \in \Sigma, y \in \Sigma^*$

$= \delta^*(\delta^*(q, y), a)$ PARTO DA q E GESTISCO y (O TERMINA O FALLISCE) E POI DA LI GESTISCO a

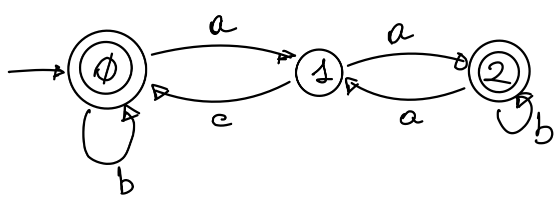
$\delta^*(q, a) = \delta(q, a)$ $\Rightarrow$ CASO BASE! NECESSARIO! (APPLICO LA * UNA SOLA VOLTA)

$$\boxed{L(\underbrace{M}_{\text{AUTOMA}}) = \{ x \in \Sigma^* \mid \delta^*(q_0, x) \in F \}}$$

IL PUNTO DI ARRIVO DEVE $\in$ AGLI STATI FINALI

ESEMPIO:

INPUT	STATO
$\boxed{a}$ c a	0
$\boxed{c}$ a	1
$\boxed{a}$	0
/	1



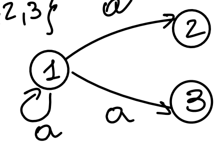
LO STATO 1 NON È FINALE QUINDI LA PAROLA <u>NON</u> APPARTIENE AL LINGUAGGIO

AUTOMI A STATI FINITI NON DETERMINISTICI

SENZA ε-MOSSE: SUGLI ARCHI C'È SEMPRE UN SIMBOLO MAI ε

$\langle Q, \Sigma, \delta, q_\phi, F \rangle$

$\delta(1,a) = \{1,2,3\}$



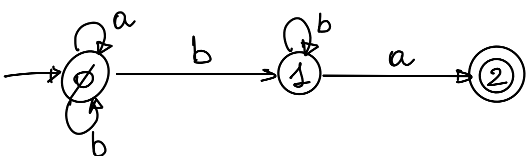
$\delta : Q \times \Sigma \rightarrow 2^Q$ → UNO DEI SOTTOINSIEMI DI Q

$\delta(q, ya)$

$\delta(q, \varepsilon) = \{q\}$

$\delta(q, ya) = \{p \mid \exists r \in \delta^*(q,y) \wedge p \in \delta(r,a)\}$

$L(M) = \{x \in \Sigma^* \mid \delta^*(q_\phi, x) \cap F \neq \Phi\}$



INPUT	STATI
bbba	$\{\phi\}$
bba	$\{0,1\}$
ba	$\{0,1\}$
a	$\{0,1\}$
/	$\{0,2\}$

ORA PUÒ ESSERE UN INSIEME DI STATI

$\delta^*(0, bbba) \rightarrow \delta^*(\delta^*(0,bbb),a) =$

$= \delta^*(\delta^*(\delta^*(\phi, bb), b), a) =$

$= \delta^*(\delta^*(\delta^*(\boxed{\delta^*(0,b)}, b), b), a)$

$\delta^*(\delta^*(\boxed{\delta^*(\{0,1\}, b)}), b), a)$

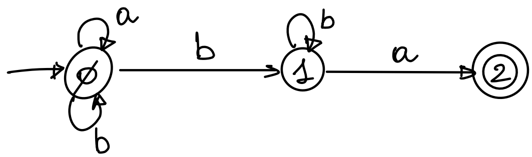
$\delta^*(\boxed{\delta^*(\{0,1\}, b)}, a)$

$\boxed{\delta^*(\{0,1\}, a)}$

$= \{0,2\}$

ALMENO UNO DEGLI STATI DEVE ESSERE FINALE

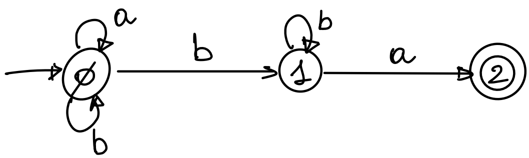
SICCOME 2 APPARTIENE AD UNA DELLE FINI DELLA COMPUTAZIONE ED È STATO FINALE LA PAROLA bbba È AL LINGUAGGIO



INPUT	STATI
$\boxed{a}aba$	$\{\emptyset\}$
$\boxed{a}ba$	$\{\emptyset\}$
$\boxed{b}a$	$\{\emptyset\}$
a	$\{\emptyset, 1\}$
/	$\{\emptyset, 2\}$

ALMENO UNO DEGLI STATI È FINALE; SICCOME 2 APPARTIENE AD UNA DELLE FINI DELLA COMPUTAZIONE ED È STATO FINALE LA PAROLA bbba È AL LINGUAGGIO (2 IN QUESTO CASO È L'UNICO STATO FINALE E VERIFICO CHE CI SIA)

$L = (a \cup b)^* b^+ a$ (LINGUAGGIO)

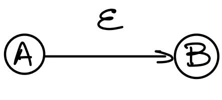


INPUT	STATI
$\boxed{b}aa$	$\{\emptyset\}$
$\boxed{a}a$	$\{\emptyset, 1\}$
$\boxed{a}$	$\{\emptyset, 2\}$
/	$\{\emptyset\}$

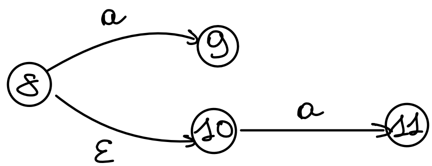
ALMENO UNO DEGLI STATI È FINALE? SICCOME L'INSIEME DI ARRIVO CONTIENE SOLO ZERO CHE <u>NON</u> È AGLI STATI FINALI, NESSUNA COMPUTAZIONE MI HA PORTATO AD UNO STATO FINALE

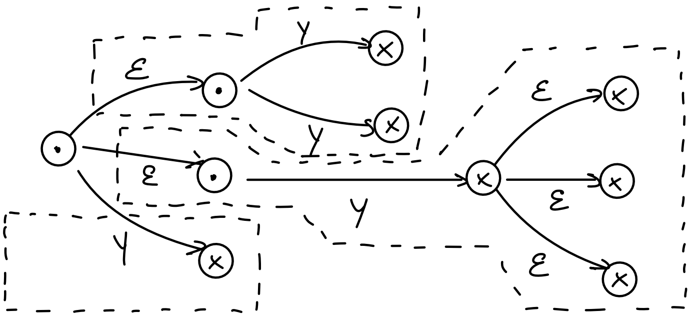
$L = (a \cup b)^* b^+ a$ (LINGUAGGIO)

CON ε-MOSSE: C'È UN ARCO TRA DUE STATI CON LABEL ε, E QUANDO LO UTILIZZO NON CONSUMO INPUT



ESEMPIO:





$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$$

ε-CHIUSURA: CIÓ CHE A PARTIRE DA UNO STATO POSSO RAGGIUNGERE A PARTIRE DA ARCHI ε

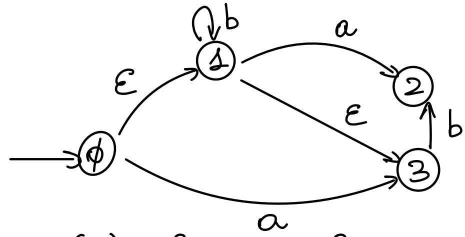
- $q \in \varepsilon\text{-CHIUSURA}(q)$
- $r \in \varepsilon\text{-CHIUSURA}(q)$ SE $\exists s \in \varepsilon\text{-CHIUSURA}(q) \land \delta(s,\varepsilon) = r$

$$\hat{\delta}(q,\varepsilon) = \varepsilon\text{-CHIUSURA}(q)$$

$$\hat{\delta}(q,ya) = \varepsilon\text{-CHIUSURA}(P) \text{ DOVE } P,$$

$$P = \{ p \mid \exists r \in \hat{\delta}(q,y) \land \delta(r,a) = p \}$$

SIMULO LA STRINGA: ab



INPUT	STATI
ab	$\{\phi\}$
b	$\{3,2\}$
/	$\{2\}$

$\varepsilon\text{-CHIUSURA}(\phi) = \{\phi, 1, 3\}$

$\delta(\varepsilon\text{-CHIUSURA}(\phi), a) = \{3,2\}$

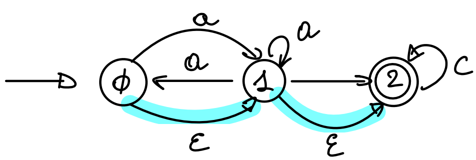
$\varepsilon\text{-CHIUS.}(\delta(\varepsilon\text{-CHIUS.}(\phi), a)) = \{3,2\} = \varepsilon\text{-CHIUS}(\{3,2\})$

$$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(q_\phi, x) \cap F \neq \Phi\}$$

$M = \langle Q, \Sigma, \delta, q_\phi, F \rangle$, VOGLIO DEFINIRE UN AUTOMA

$M' = \langle Q', \Sigma', \delta', q_\phi', F' \rangle$ PER ELIMINARE LE ε-CHIUSURE

- L'ALFABETO RIMANE INVARIATO: $\Sigma' = \Sigma$
- L'INSIEME DEGLI STATI INVARIATO: $Q' = Q$
- LO STATO INIZIALE INVARIATO: $q_\phi' = q_\phi$
- L'INSIEME DEGLI STATI FINALI PUÒ VARIARE: $F' = \begin{cases} F \\ F \cup \{q_\phi\} \text{ SE } \varepsilon\text{-CHIUSURA}(q_\phi) \cap F \neq \Phi \end{cases}$ SOLO IN UN CASO (AL PIÙ UNO AGGIUNTO, q_ϕ)

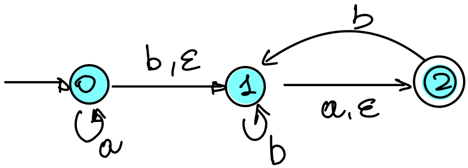


ELIMINANDO ε NON POTREI RAGGIUNGERE PIÙ 2 CHE È STATO FINALE, RIMARREI SENZA STATO FINALE (ϕ DIVENTA FINALE)

- FUNZIONE DI TRANSIZIONE: $\forall q \in Q, \forall a \in \Sigma, \delta'(q,a) = \hat{\delta}(q,a)$

$$\delta'(q,a) = \varepsilon\text{-CHIUSURA}(\delta(\varepsilon\text{-CHIUSURA}(q), a))$$

DATO IL SEGUENTE AUTOMA ELIMINARE LE ε-MOSSE



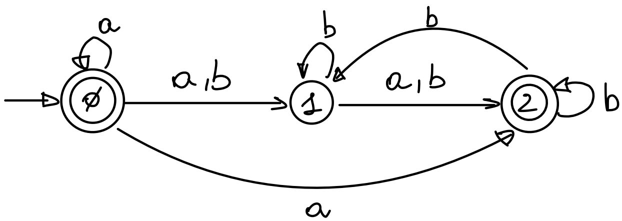
	∅	1	2
ε-CHIUSURA	∅, 1, 2	1, 2	2
δ(ε-CHI, a)	0, 2	2	/
ε-CHIUSURA(δ(ε-CHI, a))	∅, 1, 2	2	/
δ(ε-CHI., b)	1	1	1
ε-CHIUSURA(δ(ε-CHI., b))	1, 2	1, 2	1, 2

a

b

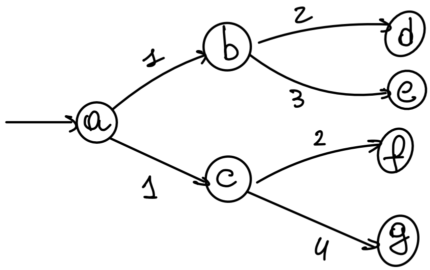
EFFETTUO QUESTI PASSAGGI PER TUTTE GLI ELEMENTI DEL MIO ALFABETO

QUESTE DUE RIGHE MI DETERMINA LA NUOVA FUNZ. DI TRANSIZIONE



AGGIUNGO ∅ AGLI STATI FINALI SE ε-CHIUSURA(∅) CONTIENE ALMENO UNO STATO FINALE → IN QUESTO CASO SI → 2

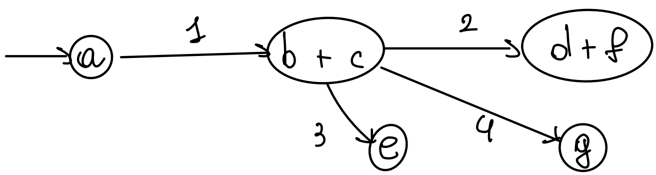
ELIMINAZIONE NON DETERMINISTICO (DA NON A DETERMINISTICO)



DEVO LAVORARE SULLA STRUTTURA DELL'AUTOMA PER TRASFORMARLO IN AUTOMA DETERMINISTICO

NEL DETERMINISMO POSSO AVERE SOLO UNA COMPUTAZIONE

$\Downarrow$



NUOVO AUTOMA DETERMINISTICO!

$M = < Q, \Sigma, \delta, q_\phi, F >$ AUTOMA NON DETERMINISTICO DI PARTENZA

VOGLIO CALCOLARE $M' = < Q', \Sigma', \delta', q_\phi', F' >$ DETERMINISTICO

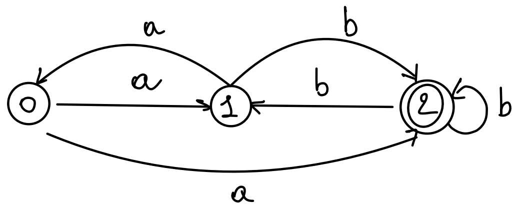
- $\Sigma' = \Sigma$ L'ALFABETO NON CAMBIA
- $Q' = 2^Q$, $Q = \{0, 2\}$ $Q' = \{\{0\}, \{2\}, \{0,2\}\}$
- $q_\phi' = [q_\phi]$ LO STATO DI PARTENZA NON CAMBIA
- $F' = \{p \in Q' \mid p \cap F \neq \emptyset\}$ SE ALMENO UNO DEGLI STATI ERA FINALE IL NUOVO STATO È FINALE (ALMENO UNO STATO FINALE PRECEDENTE)
- $\delta' : [q_1 \dots q_J] \in Q'$, $a \in \Sigma'$

SE E SOLO SE

$$\delta'(\quad , \quad) = [p_1 \dots p_K] \Longleftrightarrow \bigcup_{q = \{q_1 .. q_J\}} \delta(q, a) = \{p_1 \dots p_K\}$$

NON UN INSIEME DI STATI MA UNO UNICO CHE RAPPRESENTA UNA SITUAZIONE COMPLESSA

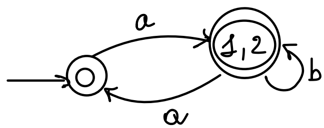
ESEMPIO

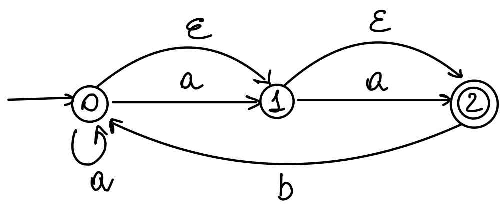


STATO FINALE

FACCIO L'UNIONE TRA a(0) E a(1)
b(0) b(1)

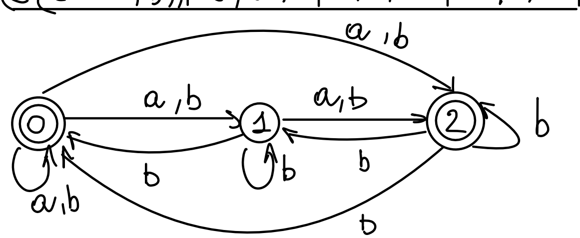
	0	1	2 *	0,1	0,2 *	1,2 *	0,1,2 *
a	1,2	0	/	0,1,2	1,2	0	0,1,2
b	/	2	1,2	2	1,2	1,2	1,2





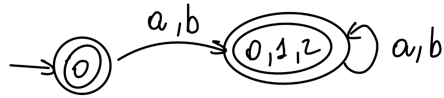
RIMUOVO LE ε-MOSSE

	0	1	2
ε-CHIUSURE	0,1,2	1,2	2
δ(ε-CHIUS, a)	0,1,2	2	/
ε-CHIUSURA(δ(ε-CHIUS, a))	0,1,2	2	/
δ(ε-CHIUS, b)	0	0	0
ε-CHIUSURA(δ(ε-CHIUS, b))	0,1,2	0,1,2	0,1,2

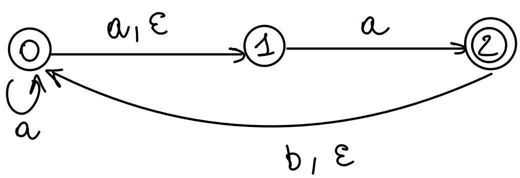


TRASFORMO IN DETERMINISTICO

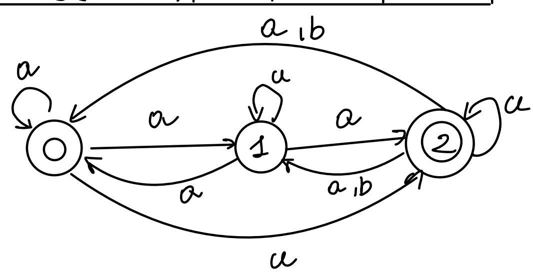
	0*	1	2*	0,1*	0,2*	1,2*	0,1,2*
a	0,1,2	2	/	0,1,2	0,1,2	2	0,1,2
b	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2



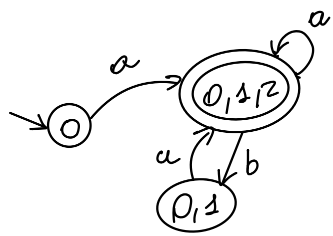
ESERCIZIO



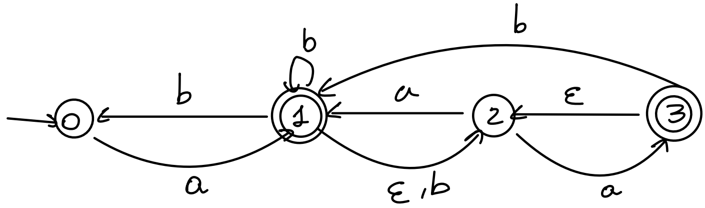
	0	1	2
ε-CHIUSURE	0,1	1	0,1,2
δ(ε-CH, a)	0,1,2	2	0,1,2
ε-CHIU δ(ε-CH, a)	0,1,2	0,1,2	0,1,2
δ(ε-CH, b)	/	/	0
ε-CHIU δ(ε-CH, a)	/	/	0,1



	0	1	2*	0,1	0,2*	1,2*	0,1,2*
a	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2
b	/	/	0,1	/	0,1	0,1	0,1

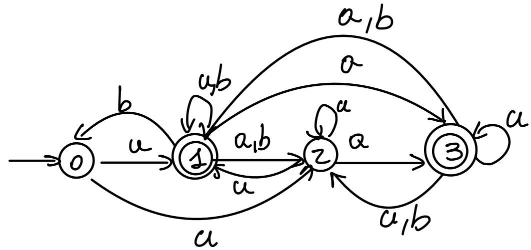


ESERCIZIO



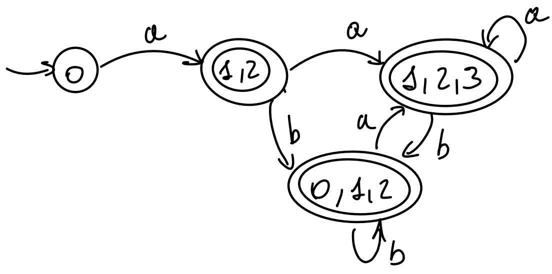
RIMUOVO ε-MOSSE

	0	1	2	3
ε-CHIUS	0	1,2	2	2,3
δ(ε-CH,a)	1	1,3	1,3	1,3
ε-CHIU(δ(ε,a))	1,2	1,2,3	1,2,3	1,2,3 ←
δ(ε-CH,b)	/	0,1,2	/	1
ε-CHIU(δ(ε-CH,b))	/	0,1,2	/	1,2 ←

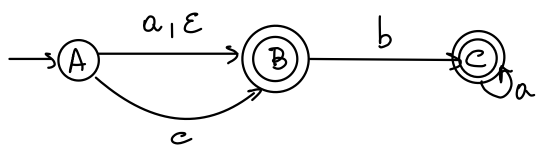


RIMUOVO NON DETERMINISMO → DETERMINISTICO

	0	1*	2	3*	0,1*	0,2	0,3*	1,2*	1,3*	2,3*	0,1,2*	0,1,3*	0,2,3*	0,1,2,3*	1,2,3*
a	1,2	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3
b	/	0,1,2	/	1,2	0,1,2	/	1,2	0,1,2	0,1,2	1,2	0,1,2	0,1,2	1,2	0,1,2	0,1,2



DA UN AUTOMA A STATI FINITI COSTRUISCO UNA GRAMMATICA LINEARE DX



AUTOMA: $< Q, \Sigma, \delta, q_\phi, F >$, GRAMMATICA: $< V, \Sigma, P, S >$

- $\forall p \in Q \quad \exists s \in V$ PER OGNI STATO ESISTE UN NON TERMINALE

$V = \{A, B, C\}$

- S (ASSIOMA) È IL PUNTO DI PARTENZA $q_\phi \rightarrow S = A$
- $\Sigma' = \Sigma$

SIMBOLO ALFABETO

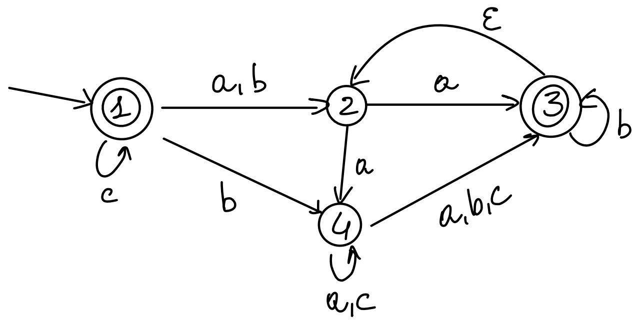
- $\forall \delta(q, a) = K$ CREO UNA REGOLA GRAMMATICALE $Q \rightarrow aK$ CON $a = \{U \mid \varepsilon\}$
  - $A \rightarrow aB$, $A \rightarrow B$, $A \rightarrow cB$
  - $B \rightarrow bC$
  - $C \rightarrow aC$

  $\Rightarrow$ QUESTA GRAMMATICA HA ALMENO UN PROBLEMA: NON VA MAI A GENERARE UNA PAROLA (*)

(*) LA TRASFORMAZIONE NON È FINITA , MANCANO GLI STATI FINALI (B, C)

- $\forall q \in F$ GENERO LA STRINGA VUOTA $q \rightarrow \varepsilon$
  - $A \rightarrow aB$, $A \rightarrow B$, $A \rightarrow cB$
  - $B \rightarrow bC$, $\boxed{B \rightarrow \varepsilon}$
  - $C \rightarrow aC$, $\boxed{C \rightarrow \varepsilon}$

ESERCIZIO



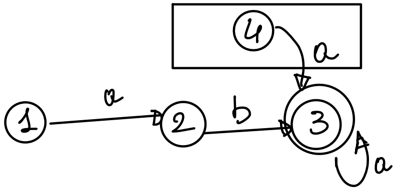
REGOLE DI PRODUZIONE:

PERCHÉ 1 È STATO FINALE

- $1 \to \varepsilon$, $1 \to a2$, $1 \to b2$, $1 \to c1$, $1 \to b4$
- $2 \to a3$, $2 \to a4$
- $3 \to \varepsilon$, $3 \to 2$, $3 \to b3$
- $4 \to a4$, $4 \to c4$, $4 \to a3$, $4 \to b3$, $4 \to c3$

AUTOMA A STATI FINITI MINIMO (DA AUTOMA DETERMINISTICO)

→ TUTTI GLI STATI DEVONO ESSERE RAGGIUNGIBILI DALLO START NODE



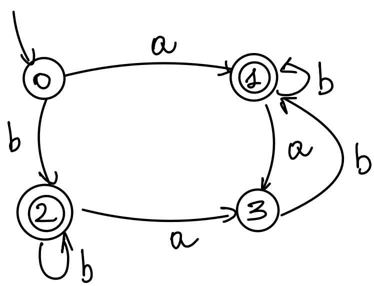
NODO NON RAGGIUNGIBILE DALLO START NODE

→ DEVO POTER SEMPRE RAGGIUNGERE UN NODO FINALE ANCHE SE NON FALLISCO (DIVENTEREBBE UNA STRADA INUTILE, RISPOSTA → NO)

LE SITUAZIONI INUTILI PRIMA VANNO RIMOSSE, POI APPLICO L'ALGORITMO DI MINIMIZZAZIONE (PERCHÉ LUI NON LE ELIMINA)

- FONDO GLI STATI CHE FANNO LE STESSE COSE (ALGORITMO MINIMO)
  - CONTROLLO SUL FINALE: ① ②, NON POSSO FONDERLI PERCHÉ NON SONO FINALI O ENTRAMBI NON FINALI
  - CONTROLLO EQUIVALENZA STATI: TRAMITE LA FUNZ. TRANSIZIONE PROPRIETÀ TRANSITIVA E SIMMETRICA

ESEMPIO:



$\emptyset \equiv 1$? , $\emptyset \equiv 3$? $\emptyset \equiv 2$?

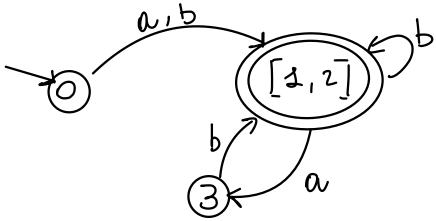
SE $\emptyset \equiv 1$ E $1 \equiv 2 \Rightarrow \emptyset \equiv 2$

UTILIZZO UNA MATRICE DIAGONALE DI SUPPORTO

	0	1	2
1	X		
2	X	(3,3) (1,2)	
3	X	X	X

(1,/)

1. CONTROLLO SUBITO SUL FINALE
2. ∀ SIMBOLO DELLA FUNZ. TRANSIZIONE FACCIO DELLE CONSIDERAZIONI (SEGNO L'OUTPUT DEL NODO CON L'ARCO)



HO UNITO GLI STATI 1,2 COME TRADUZIONE DELLA MATRICE, ORA DEVO RICALCOLARE LA FUNZIONE DI TRANSIZIONE SU 3 STATI E NON PIÙ 4

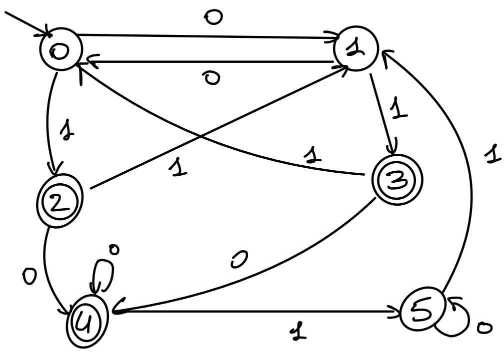
$\forall q \in Q \quad \forall a \in \Sigma$

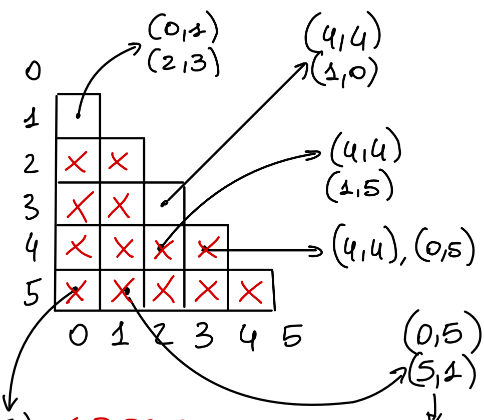
NUOVA FUNZ. TRANSIZIONE

IF $\delta(q, a) = s$ THEN $\delta([q], a) = [s]$

L'AUTOMA MINIMO NON È UNICO!

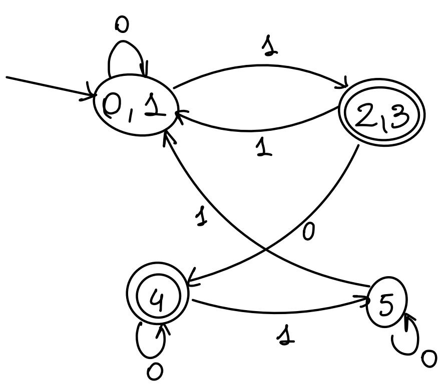
ESERCIZIO





DOPO L'ANALISI POSSO DIRE CHE: $0 \equiv 1$ E $2 \equiv 3$

(1,5) X PERCHÉ LA CELLA
(2,1) (2,1) CONTIENE X (NON SONO ≡)



AUTOMA MINIMO!

A PARTIRE DA UN AUTOMA A STATI FINITI COSTRUISCO IL NEGATO CHE MI RICONOSCE IL COMPLEMENTO (LINGUAGGIO COMPLEMENTARE)

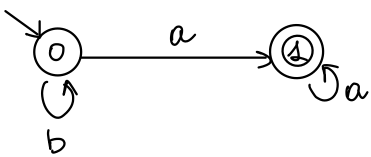
$$M = \langle Q, \Sigma, \delta, q_\phi, F \rangle$$

$$\overline{M} = \langle \overline{Q}, \Sigma, \overline{\delta}, \overline{q_\phi}, \overline{F} \rangle$$

- $\overline{q_\phi} = q_\phi$
- $\overline{Q} = Q \cup \{P\}$ → GESTISCE DELLE SITUAZIONI (AGGIUNTA DI UNO STATO) → "POZZO", "BUCO NERO"
- $\overline{F} = (Q - F) \cup \{P\}$ (SWITCH RISPETTO AI PRECEDENTI FINALI)
- SE $\forall q \in Q, \forall a \in \Sigma \;\; \delta(q,a) = K$ ALLORA $\overline{\delta}(q,a) = K$

  SE $\forall q \in Q, \forall a \in \Sigma \;\; \delta(q,a) = /$ ALLORA $\overline{\delta}(q,a) = P$ ✱

  $\forall q \in Q, \forall a \in \Sigma \;\; \delta(P,a) = P$ ✱



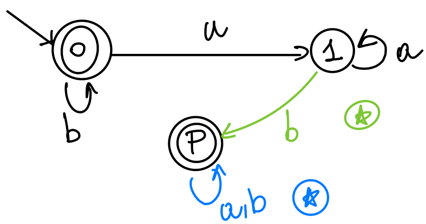
(MI RICONOSCE $b^* a^+$)

aaa → SI!

bb → NO!

bab → NO!

AUTOMA COMPLEMENTARE



aaa → NO!

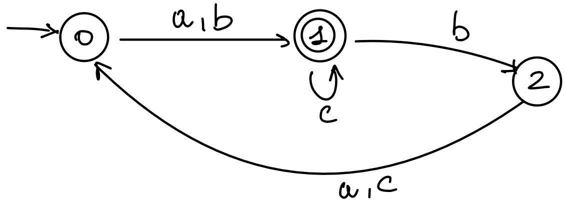
bb → SI!

bab → NO!

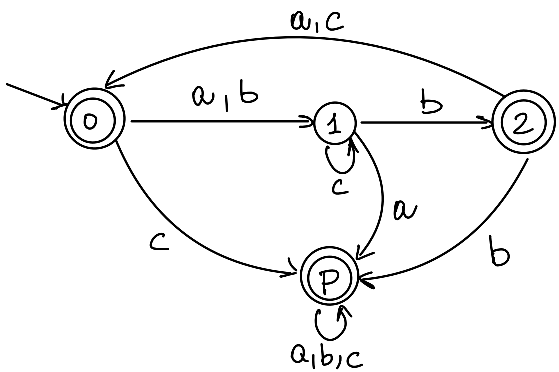
DEVONO AVERE COMPORTAMENTI OPPOSTI ✱

ESERCIZIO

AUTOMA INIZIALE



AUTOMA COMPLEMENTARE

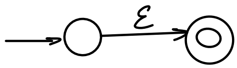


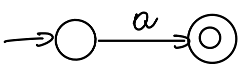
SE L_1, L_2 SONO LINGUAGGI REGOLARI $L_1 \cap L_2$ (INTERSEZIONE) IL LINGUAGGIO RISULTANTE É ANCORA REGOLARE?

OPPURE DUE ESPRESSIONI REGOLARI $e_1 \cap e_2$ (INTERSEZIONE) L'ESPRESSIONE RISULTANTE É REGOLARE?

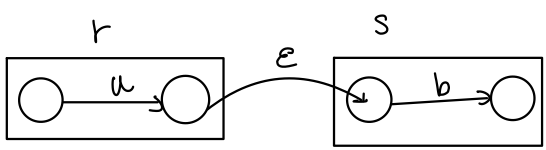
$\Rightarrow \quad e_1 \cap e_2 \Rightarrow \overline{\overline{e_1} \cup \overline{e_2}}$ $\forall$ ESPRESS. REGOLARE C'E UN AUTOMA A STATI FINITI, DI CUI POSSO FARE IL COMPLEMENTO (NEGATO) QUINDI SE POI ESEGUO L'UNIONE AVRÒ ANCORA UN'ESPRESSIONE REGOLARE, EFFETTUO ANCORA IL NEGATO (AUTOMA COMPLEMENTARE)

ESPRESSIONE REGOLARE ⇒ AUTOMA STATI FINITI NON DETERMINISTICO (THOMPSON)

- $\Phi \Rightarrow$  (QUALSIASI INPUT DICO NO!)
- $\varepsilon \Rightarrow$ 

- $a \quad \forall a \in \Sigma$ 

- r, s ESP. REG → $r \cdot s$ È ESP. REG.

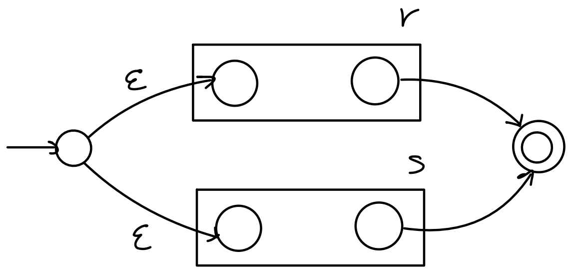


DALL'ULTIMO STATO DI r ALLO STATO INIZIALE DI S

ES.: $a \cdot b$

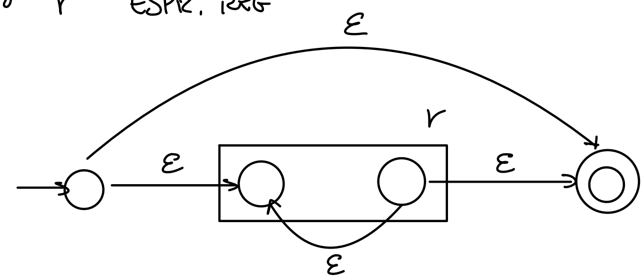
LO STATO FINALE DIVENTA IL FINALE DI S

- r, s ESP. REG. → $r \cup s$ ESP. REG.

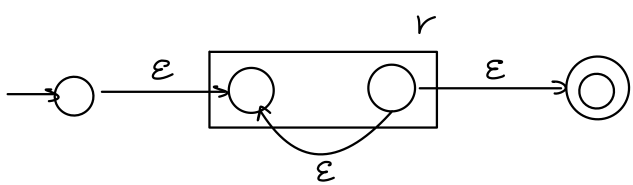


AGGIUNGO UNO START NODE E LO COLLEGO CON ε A r, s ED INFINE AGGIUNGO UN NODO FINALE

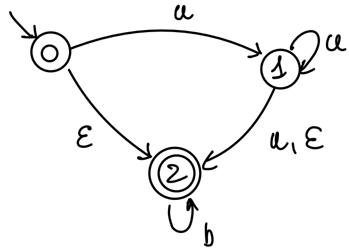
- r^* ESPR. REG



- r^+ ESPR. REG:

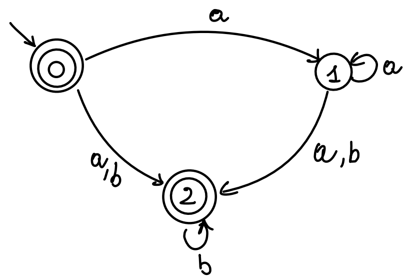


ESERCIZIO



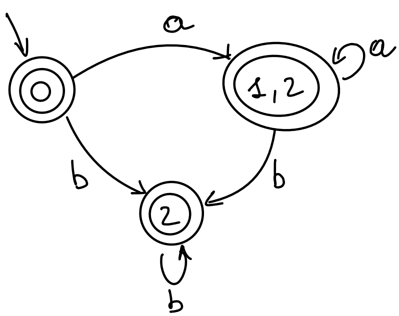
ELIMINO LE ε-MOSSE

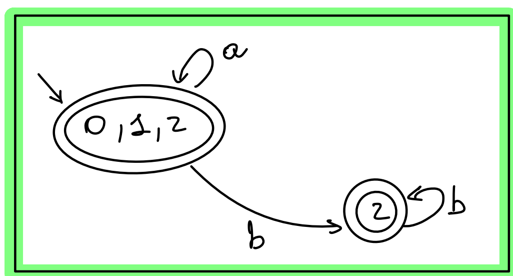
	0	1	2
ε-CHIUSURA	0,2	1,2	2
δ(ε-CHIUS, a)	1	1,2	/
→ ε-CHIUSURA(δ(ε-CHI, a))	1,2	1,2	/
δ(ε-CHIUS, b)	2	2	2
→ ε-CHIUSURA(δ(ε-CHIU, b))	2	2	2



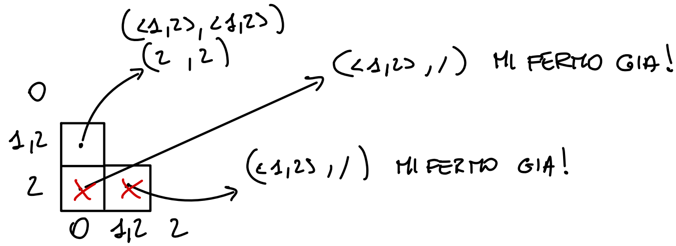
RIMUOVO IL NON-DETERMINISMO!

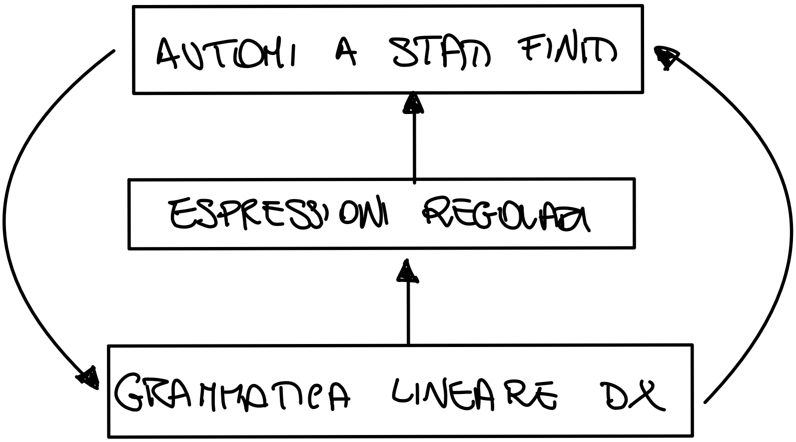
	→ 0*	1	2*	0,1*	0,2*	1,2*	0,1,2*
a	1,2	1,2	/	1,2	1,2	1,2	1,2
b	2	2	2	2	2	2	2





TROVO L'AUTOMA MINIMO!





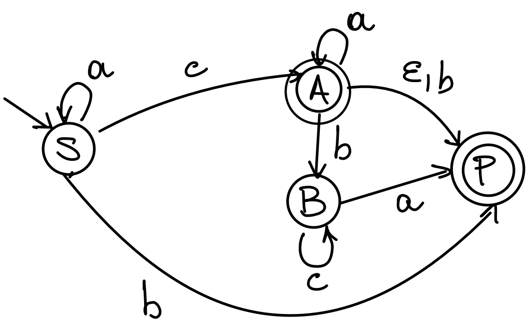
DALLA SEGUENTE GRAMMATICA DX COSTRUIRE L'AUTOMA A STATI FINITI EQUIVALENTE IN MODO AUTOMATICO

$S \rightarrow aS \mid b \mid cA$

$A \rightarrow \varepsilon \mid b \mid bB \mid aA$

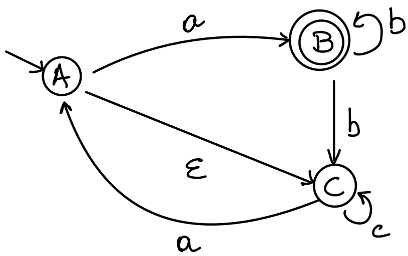
$B \rightarrow cB \mid a$

- ∀ NON TERMINALE HO UN NODO, PIÙ IL NODO POZZO P (FINALE X COSTRUZIONE)
- LO START NODE È L'ASSIOMA (S) (SEMPRE S SE NON DIVERSAMENTE SPECIFICATO)



⊛ QUALUNQUE REGOLA CHE PRODUCA ε DIVENTERÀ UNO STATO FINALE

DALL'AUTOMA A STATI FINITI ALLA GRAMMATICA LINEARE DX



⟹

$A \rightarrow aB \mid C$

$B \rightarrow bB \mid bC \mid \varepsilon$

$C \rightarrow cC \mid aA$

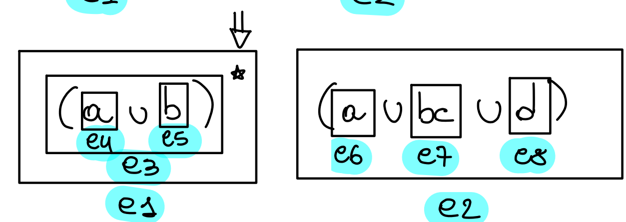
GESTISCE LO STATO FINALE B

DALL'ESPRESSIONE REGOLARE CREARE UN AUTOMA A STATI FINITI SENZA CREARE LA GRAMMATICA

$(a \cup b)^*$ e1 $\quad (a \cup bc \cup d)$ e2

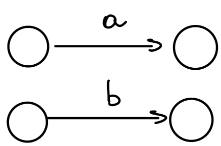
- DIVIDO L'ESPR. REGOLARE IN SOTTO ESPR. REGOLARI PIÙ PICCOLE FINO AD ARRIVARE AI CASI BASE

PARTO DALL'ESPR. PIÙ INTERNE E RICOSTRUISCO A SALIRE L'ESP. REGO.

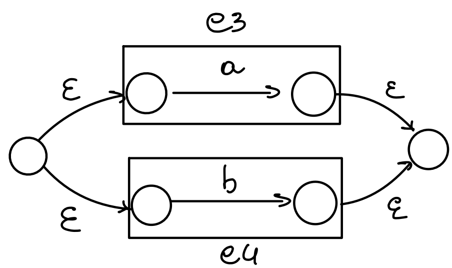


e3 :

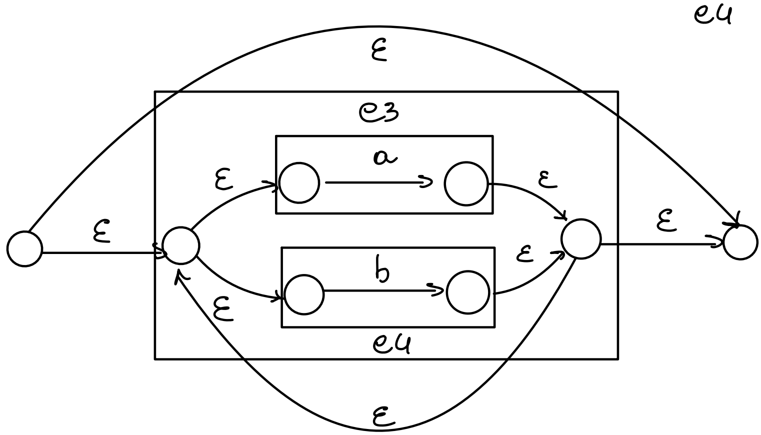
e4 :



⟹ e3 :

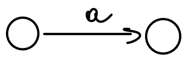


⟹ e1 :

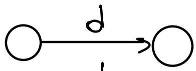


⟹

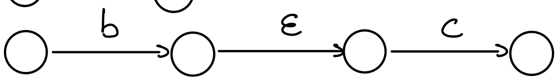
e6 :



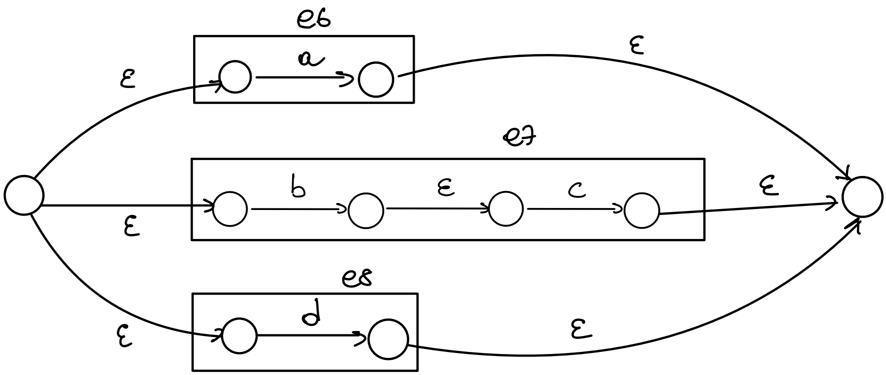
e8 :

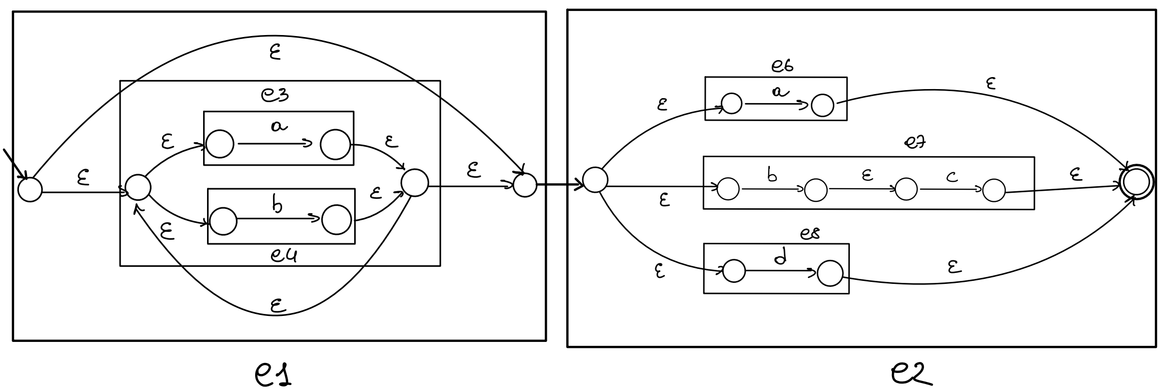


e7 :



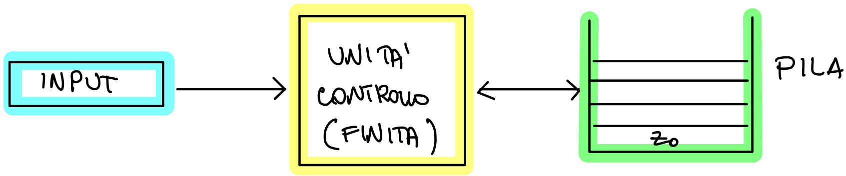
e2 :

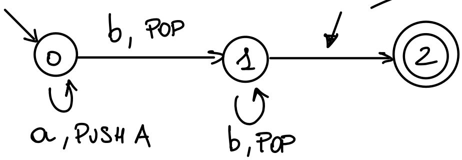




LO STATO FINALE SARA LO L'ULTIMO STATO DI e2 MENTRE QUELLO INIZIALE SARA' LO STATO INIZIALE DI e1

AUTOMI A PILA





→ SIMBOLO DI FINE LETTURA INPUT

QUESTO AUTOMA RICONOSCE IL LINGUAGGIO:

$a^n b^n$ CON $n \geq 1$

PILA	INPUT	STATO
Z_0	a a b b ↙	$\emptyset$
Z_0, A	a b b ↙	$\emptyset$
Z_0 A, A	b b ↙	$\emptyset$
Z_0, A	b ↙	1
Z_0	↙	1
Z_0		2

① NON PERMETTERÒ MAI DI FARE UN'OPERAZIONE DI POP SE LA PILA È VUOTA → UTILIZZERÒ Z_0 CHE RAPPRESENTA IL "BOTTOM" DELLA PILA

POSSO DIRE "SI" SOLO QUANDO HO CONSUMATO TUTTO L'INPUT (E SONO STATO FIN.)

DEFINIZIONE

$$\langle Q, \Sigma, \tau, \delta, q_0, Z_0, F \rangle$$

- Q : INSIEME DI STATI
- Σ : ALFABETO DI INPUT
- τ : (TAU) ALFABETO DELLA PILA (SIMBOLI CHE POSSO METTERE IN PILA)
- δ : FUNZIONE DI TRANSIZIONE
- q_0 : STATO INIZIALE
- Z_0 : SIMBOLO ALFABETO DELLA PILA → RAPPRESENTA UNA PILA VUOTA, TUTTO QUELLO CHE STA SOTTO NON VIENE CONSIDERATO
- F : INSIEME STATI FINALI

$$\delta : (Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \longrightarrow 2^{Q \times \Gamma^*}$$ (FUNZ. TRANSIZIONE NON DETERMINISTICA)

E.S.: $\delta(q_0, a, A) = (q_0, AA)$

VIENE SEMPRE CONSUMATO (POP) – CONSUMO LA TESTA DELLA PILA

POSSO AVERE DUE TIPI DI REGOLE

$$\delta(q, a, x) = \{(q_1, \alpha_1) \dots (q_n, \alpha_n)\}$$

$$\delta(q, \varepsilon, x) = \{(q_1, \alpha_1) \dots (q_n, \alpha_n)\}$$

CON $\alpha_1, \dots, \alpha_n \in \Gamma$

UN AUTOMA NON DETERMINISTICO A PILA <u>NON</u> PUÒ ESSERE RIDOTTO AD UN AUTOMA DETERMINISTICO A PILA

$$\delta(q, \boxed{a}, X) = (x, y) \qquad x \in \Gamma, \; y \in \Gamma^*$$

CONFIGURAZIONE

$$(q, q, \gamma) \in Q \times \Sigma^* \times \Gamma^*$$

NELL'AUTOMA A PILA DATA UNA CONFIGURAZIONE VIENE DEFINITA MOSSA UN PASSAGGIO AD UN'ALTRA CONFIGURAZIONE (TRAMITE FUNZ. TRANSIZI.)

DUE CASI:

- $(q, aY, \gamma X) \longrightarrow (p, Y, \gamma\pi)$ SE $(p, \pi) \in \delta(q, a, X)$ — $\pi \in \Gamma^*$

  q: STATO; aY: STRINGA INPUT RIMANENTE; γX: PILA, $\gamma \in \Gamma^*$

- $(q, Y, \gamma X) \longrightarrow (p, \gamma\pi)$ SE $(p, \pi) \in \delta(q, \varepsilon, X)$

RICONOSCIMENTO PER AUTOMI A PILA

CONFIGURAZIONE INIZIALE: (q_0, A, Z_0)

1- RICONOSCIMENTO PER STATO FINALE: UNA VOLTA CONCLUSO L'INPUT ARRIVO IN UNO STATO FINALE

$$(q_0, A, Z_0) \xrightarrow{*} (q, \varepsilon, \gamma) \text{ con } q \in F$$

INDIPENDENTEMENTE DALLA PILA POSSO DIRE SI / NO

2- RICONOSCIMENTO PER PILA VUOTA: QUANDO FINISCO L'INPUT E SONO IN UNO STATO E LA PILA È VUOTA (NO Z_0)

$$(q_0, A, Z_0) \xrightarrow{*} (q, \varepsilon, \varepsilon) \quad q \in Q \text{ (ANCHE NON FINALE)}$$

AUTOMA A PILA DETERMINISTICO

VINCOLI SULLA FUNZ. TRANSIZIONE ATTI A RENDERE POSSIBILE AL MASSIMO UNA MOSSA.

È POSSIBILE AVERE MOSSE CHE NON CONSUMANO INPUT (ε-MOSSE)

TRE VINCOLI SULLA FUNZ. TRANSIZIONE:

1- $|\delta(q, a, x)| \leq 1$

2- NON DEVE SUCCEDERE CHE (SAREBBE NON DETERMINI.) $\begin{cases} \delta(q, a, x) \\ \delta(q, \varepsilon, x) \end{cases}$ NON DEVONO ESSERE DEFINITI CONTEMPORANEAMENTE

3- $|\delta(q, \varepsilon, x)| \leq 1$

GLI AUTOMI A PILA DETERMINISTICI RICONOSCONO MENO DEGLI AUTOMI A PILA NON DETERMINISTICI OVVERO RICONOSCONO UN SOTTOINSIEME DI UN LINGUAGGIO CONTEXT FREE

AUTOMI A PILA NON DETERMINISTICA

UTILIZZATI PER QUALSIASI LINGUAGGIO CONTEXT FREE

DA GRAMMATICA CONTEXT FREE A AUTOMA A PILA NON DETERMINISTICO

$< V, \Sigma, P, S >$ GRAMMATICA CONTEXT FREE

$< Q, \Sigma, \Gamma, \delta, q_0, Z_0, F >$ AUTOMA A PILA

È SIMILE ALLA LOGICA PER LA COSTRUZIONE AUTOMATICA DEGLI AUTOMI A STATI FINITI DA UNA GRAMMATICA LINEARE DX.
IDEA → TENTO DI SIMULARE LA GRAMMATICA (FUNZ. TRANSIZIONE) CON L'AUSILIO DELLA PILA

L'AUTOMA COSÌ COSTRUITO FA ACCETTAZIONE SOLO SU PILA VUOTA PERCHÉ F È DEFINITO COME INSIEME VUOTO $< Q, \Sigma, \Gamma, \delta, q_0, Z_0, / >$

$Q = \{ q_0 \}$ C'È SOLO LO START NODE

$\Sigma = \Sigma$ L'ALFABETO RIMANE INVARIATO

Z_0 RIMANE IL SIMBOLO PER IL FONDO DELLA PILA

$\Gamma = \{ Z_0 \} \cup \Sigma \cup V$ → HO TANTI SIMBOLI SULLA PILA QUANTI SONO QUELLI DELL'ALFABETO E I NON TERMINALI

FUNZIONE DI TRANSIZIONE:

1- CONSIDERO LE REGOLE DENTRO P, $\forall A \rightarrow \alpha$ E LE POSSO DIVIDERE IN DUE CATEGORIE A SECONDA SE α È UN TERMINALE O NON TERMINALE

- 1- $\alpha = a \beta$
- 2- $\alpha = X \beta$

CON β SEQUENZA DI TERMINALI / NON TERMINALI

$\delta(q_0, a, A) = (q_0, \beta^R)$ → β REVERSE

$\delta(q_0, \varepsilon, A) = (q_0, \beta^R X)$

2- BASATO SULL'ALFABETO: $\forall a \in \Sigma$

$$\delta(q_0, a, a) = (q_0, \varepsilon)$$

CONSUMO IL SIMBOLO SIA DALL'INPUT SIA DALLA TESTA DELLA PILA

3- REGOLE DELLA FUNZ. TRANSIZIONE: INIZIALIZZAZIONE E TERMINAZIONE

- INIZIALIZZAZIONE: $\delta(q_0, \varepsilon, Z_0) = (q_0, \swarrow S)$ SCRIVO FINE STRINGA E ASSIOMA
- TERMINAZIONE: $\delta(q_0, \swarrow, \swarrow) = (q_0, \varepsilon)$

ESERCIZIO:

$S \rightarrow aS$
$S \rightarrow A$
$A \rightarrow aAb$
$A \rightarrow ab$

$L = \{a^m b^n\}$ CON $m \geq n \geq 1$

É PRESENTE UN BILANCIAMENTO CHE LEGA a CON b

REGOLE DI PRODUZIONE:

- REGOLE CHE INIZIANO CON UN TERMINALE:

① $S \rightarrow aS \Rightarrow \delta(q_0, a, S) = (q_0, \overset{\beta^R}{S})$

② $A \rightarrow aAb \Rightarrow \delta(q_0, a, A) = (q_0, \overset{\beta^R}{bA})$

③ $A \rightarrow ab \Rightarrow \delta(q_0, a, A) = (q_0, \overset{\beta^R}{b})$

- REGOLE CHE INIZIANO CON UN NON TERMINALE:

④ $S \rightarrow A \Rightarrow \delta(q_0, \varepsilon, S) = (q_0, A)$ NON C'È β^R IN QUESTO CASO

- $\forall a \in \Sigma \Rightarrow \delta(q_0, a, a) = (q_0, \varepsilon)$:

⑤ $a \Rightarrow \delta(q_0, a, a) = (q_0, \varepsilon)$

⑥ $b \Rightarrow \delta(q_0, b, b) = (q_0, \varepsilon)$

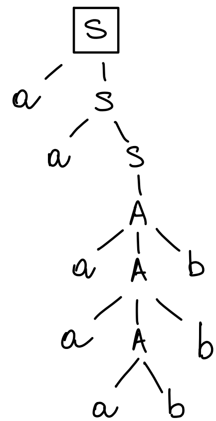
- REGOLE DELLA FUNZIO. TRANSIZIONE (INIZIALIZZAZ., TERMINAZIONE)

  - INIZIALIZZAZIONE: ⑦ $\delta(q_0, \varepsilon, Z_0) = (q_0, \swarrow S)$

  - TERMINAZIONE: ⑧ $\delta(q_0, \swarrow, \swarrow) = (q_0, \varepsilon)$

PILA	INPUT	STATO	MOSSA
Z_0	aaaaabbb$\swarrow$	q_0	7
$\swarrow$, S	aaaaabbb$\swarrow$	q_0	1
$\swarrow$, S	aaaabbb$\swarrow$	q_0	1
$\swarrow$, S	aaabbb$\swarrow$	q_0	4
$\swarrow$, A	aaabbb$\swarrow$	q_0	2
$\swarrow$, b, A	aabbb$\swarrow$	q_0	2
$\swarrow$, b, b, A	abbb$\swarrow$	q_0	3
$\swarrow$, b, b, b	bbb$\swarrow$	q_0	6
$\swarrow$, b, b	bb$\swarrow$	q_0	6
$\swarrow$, b	b$\swarrow$	q_0	6
$\swarrow$	$\swarrow$	q_0	8
/	/	q_0	$\rightarrow$ SI!

OGNI VOLTA CHE HO IL NON DETERMINISMO UTILIZZO L'ALBERO DI DERIVAZIONE PER SCEGLIERE



TRACCIA DI REGOLE

① ① ④ ② ② ③

L'AUTOMA RISPONDE SI PERCHÉ LA PILA È VUOTA, L'INPUT È VUOTO ED È NELLO STATO q_0 $\Rightarrow$ RISPOSTA: SI!

PARSER

SONO AUTOMI CHE MENTRE FANNO ACCETTAZIONE, TENTANO DI COSTRUIRE IN PARALLELO L'ALBERO DI DERIVAZIONE QUINDI LA CONSEGUENTE INTERPRETAZIONE DELL' INPUT.
(SE L'INPUT NON APPARTIENE AL LINGUAGGIO NON SARÀ NEANCHE POSSIBILE FARE INTERPRETAZIONE ⇒ LAVORO SPRECATO)

PER COSTRUIRE L'ALBERO DI DERIVAZIONE, IN BASE A DOVE INIZIO A COSTRUIRE L'ALBERO HO DUE APPROCCI

- TOP - DOWN: PARTO DALLA ROOT
- BOTTOM UP: PARTO DALLA PAROLA E ARRIVO ALLA ROOT ⊛

NOI VEDREMO GLI APPROCCI BOTTOM UP: LR(K)

- K: OLTRE AL SIMBOLO CHE CONSUMO POSSO GUARDARE ALTRI SIMBOLI DELL' INPUT (IN AVANTI) SENZA CONSUMARLI, OTTENGO INFORMAZIONI

- L: LEFT, L'INPUT VIENE LETTO DA SINISTRA

- R: RIGHT - MOST: CERCA DI IDENTIFICARE LA REGOLA DI RIDUZIONE PIÙ A DESTRA POSSIBILE

(POSSO AVERE SIA PER L CHE PER R POLITICHE DIVERSE: LEFT-MOST, ...)

- LR(0): NON VA A GUARDARE I CARATTERI SUCCESSIVI DELL' INPUT ($K = 0$)
- SLR(1): SIMPLE LR(1) GUARDA 1 CARATTERE SUCCESSIVO DELL' INPUT ($K=1$)

PER ENTRAMBE LE TIPOLOGIE GLI AUTOMI USATI SONO DETERMINISTICI E SI BASANO SU AUTOMI A PILA, PILA SULLA QUALE SCRIVONO E LEGGONO.
SI CHIAMANO ANCHE PARSER SHIFTER REDUCE: SONO POSSIBILI 4 COMANDI: SHIFT, REDUCE, ERROR, ACCEPT
SHIFT: CONSUMA UN CARATTERE SULL'INPUT
REDUCE: NON CONSUMA INPUT, LAVORO SOLO SULLA PILA

PARTE RIDUCIBILE DI UNA FORMA SENTENZIALE:

DATA UNA FORMA SENTENZIALE γ (GAMMA) SE ESISTE UNA REG. PRODUZIONE

1- $\overset{\gamma}{\boxed{A \to \beta}}$, DEVE ESISTERE PARTENDO DA UN ASSIOMA S

2- $S \underset{rm}{\overset{*}{\Longrightarrow}} \alpha A w$ (w: SEQUENZA DI TERMINALI ANCHE VUOTA)

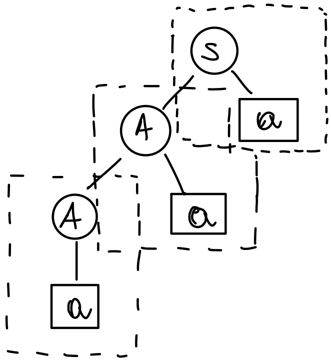
(rm: RIGHT-MOST)

$\underset{rm}{\Longrightarrow} \overset{\gamma}{\boxed{\alpha \beta w}}$ (A: NON TERMINALE)

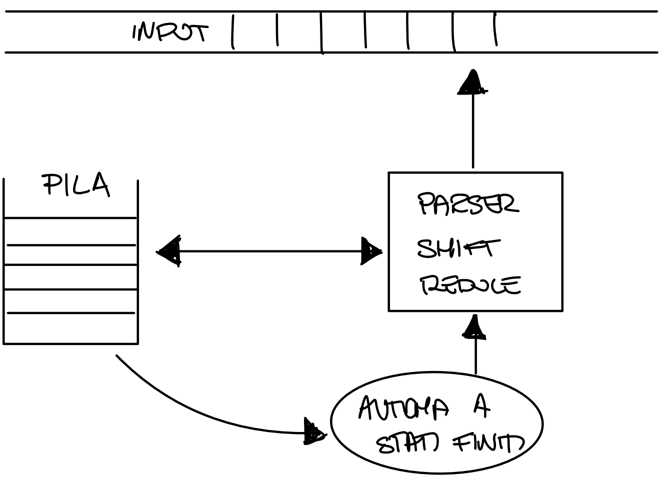
(α:

ES.: PAROLA → aaa

ALBERO DI DERIVAZIONE



LR(0)



RISPETTO AGLI AUTOMI VISTI FINO AD ORA CI SONO DUE GROSSE DIFFERENZE

- LE LABEL POSSONO CONTENERE ANCHE NON TERMINALI
- ALL'INTERNO DI OGNI STATO (OLTRE AL NOME DELLO STATO) HO DEI CANDIDATI ALLE PARTI RIDUCIBILI CHE AIUTANO AD IDENTIFICARE LA RIDUZIONE RIGHT-MOST

COS'È UN CANDIDATO RIDUCIBILE PER LR(0)?

$X \rightarrow x_1, x_2, x_3$ CON $x_1, x_2, x_3 \in (V \cup \Sigma)$ POSSONO ESSERE TERMINALI O NON TERMINALI

$X \rightarrow {}^{\triangledown}x_1\, x_2\, x_3$

$X \rightarrow x_1{}^{\triangledown}\, x_2\, x_3$

$X \rightarrow x_1\, x_2{}^{\triangledown}\, x_3$

$X \rightarrow x_1\, x_2\, x_3{}^{\triangledown}$

$\triangledown \rightarrow$ LEGGO QUALCOSA CHE C'È A SX E MI ASPETTO DI TROVARE QUELLO CHE C'È A DX

DEFINIAMO IL CONTENUTO DI OGNI STATO; È UN INSIEME DI CANDIDATI, ALMENO UNO I(STATO), DOVE I È UN NON TERMINALE E MI DEVO CALCOLARE TUTTI I SUOI CANDIDATI ATTRAVERSO LA CLOSURE(I)

- $\forall$ CANDIDATO $i \in I$ $\quad i \in$ CHIUSURA(I) (INIZIALIZZAZIONE)
- $\forall$ $A \rightarrow \alpha {}^{\triangledown} B\beta \in$ CHIUSURA(I), REGOLE CHE A DX DI $\triangledown$ PRESENTANO UN NON TERMINALE

  $\forall$ $B \rightarrow \gamma \in G$

  $B \rightarrow {}^{\triangledown}\gamma \in$ CHIUSURA(I)

SEMPLICEMENTE DEVO AVERE UNA REGOLA! $S' \rightarrow S\cancel{\$}$

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid n$

CHIUSURA $(\{E \to {}^{\triangledown} E + T\})$

$E \to {}^{\triangledown} E + T$
$E \to {}^{\triangledown} T$
$T \to {}^{\triangledown} T * F$
$T \to {}^{\triangledown} F$
$F \to {}^{\triangledown} (E)$
$F \to {}^{\triangledown} n$

ENTRAMBI I CANDIDATI SONO TERMINALI STOP

TUTTI I CANDIDATI

DEFINIZIONE OPERAZIONE GOTO

GOTO (I, x) : MI PERMETTE DI SPOSTARMI DA UNO STATO AD UN'ALTRO ALL'INTERNO DI UN AUTOMA CONSUMANDO UN TERMINALE O UN NON TER.

$\forall A \to \alpha^{\triangledown} x \beta \in I$

CHIUSURA $(\{A \to \alpha X^{\triangledown} \beta\})$ E GOTO (I, x)

PUÓ ESSERE UN NON TERMINALE O UN TERMINALE

SIMULAZIONE : COSTRUZIONE AUTOMA LR0 DALLA GRAMMATICA

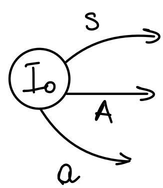
$S' \to S \swarrow$
$S \to SA \mid A$
$A \to aSb \mid ab$

I_0

$S' \to {}^{\triangledown} S \swarrow$
$S \to {}^{\triangledown} SA$
$S \to {}^{\triangledown} A$
$A \to {}^{\triangledown} aSb$
$A \to {}^{\triangledown} ab$

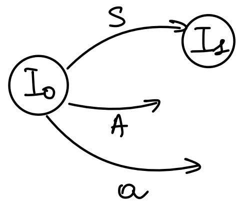
CANDIDATI

PARTO DALL'INIZIO (STATO INIZIALE)
CALCOLO LA CHIUSURA $(\{S' \to {}^{\triangledown} S \swarrow\})$

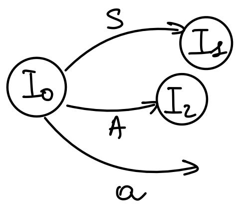


I_1

$S' \to S^{\triangledown}$
$S \to S^{\triangledown} A$
$A \to {}^{\triangledown} aSb$
$A \to {}^{\triangledown} ab$

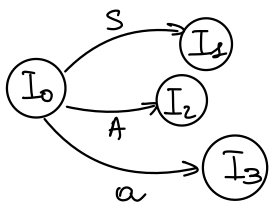


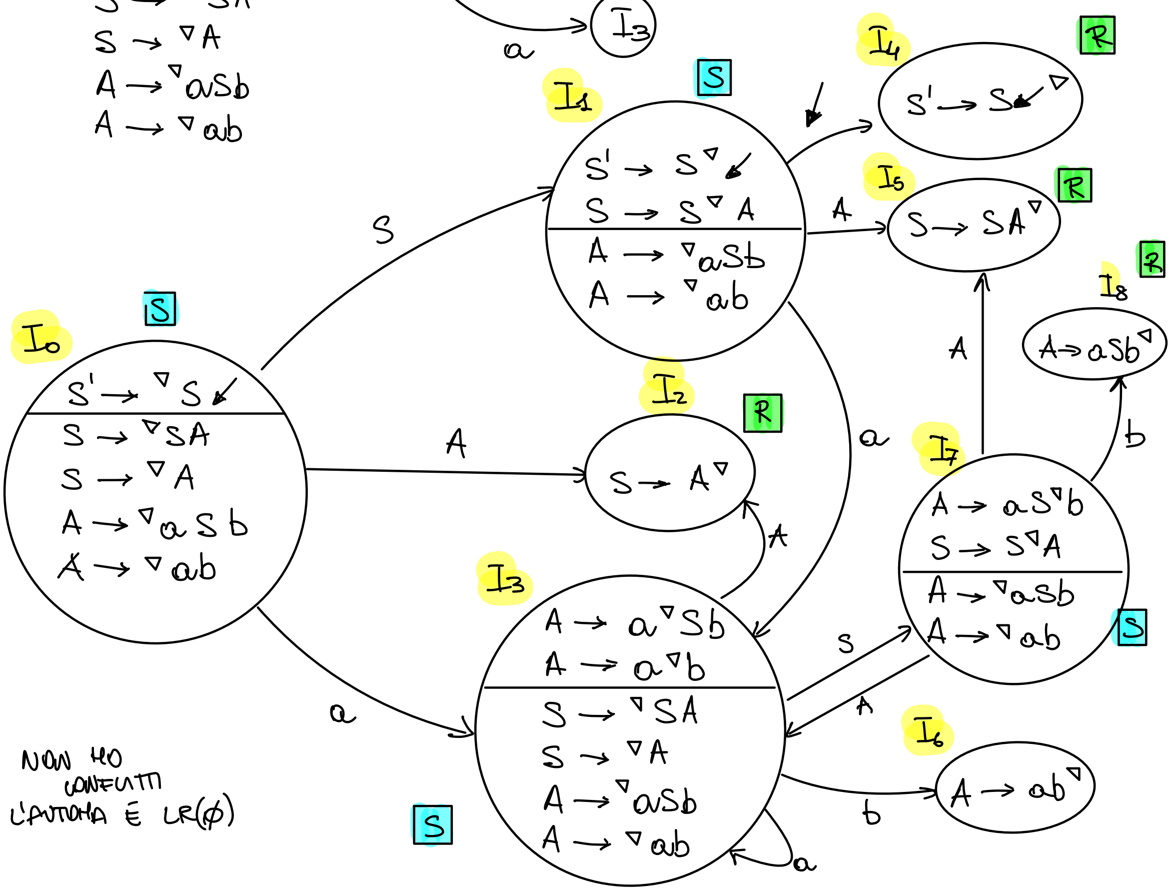
I_2 $S \to A^{\triangledown}$



I_3

$A \to a^{\triangledown} Sb$
$A \to a^{\triangledown} b$
$S \to {}^{\triangledown} SA$
$S \to {}^{\triangledown} A$
$A \to {}^{\triangledown} aSb$
$A \to {}^{\triangledown} ab$





NON HO CONFLITTI
L'AUTOMA È LR($\emptyset$)

ORA DEVO CONTROLLARE GLI STATI DI REDUCE: I_4, I_5, I_8, I_2; STATI IN CUI TROVO CANDIDATI CON IL $\triangledown$ TUTTO A DX

POI CONTROLLO GLI STATI DI SHIFT: I_0, I_3, I_7, I_1; STATI IN CUI TROVO IL $\triangledown$ E A DX HO ALTRO (NON ALLA FINE)

TIPOLOGIE DI PROBLEMI

- REDUCE / REDUCE: HO DUE CANDIDATI DI RIDUZIONE E NON POSSO SCEGLIERE DIVENTA NON DETERMINISTICO

$$A \to ab^{\triangledown}$$
$$B \to ab^{\triangledown}$$

- SHIFT / REDUCE: HO DUE CANDIDATI CHE MI DICONO L'OPPOSTO SHIFT/REDUCE

$$A \to a^{\triangledown} b A b$$
$$A \to a^{\triangledown}$$

SE SI PRESENTANO NON È UN AUTOMA LR(∅), IL LINGUAGGIO NON È TRATTABILE CON QUESTO AUTOMA (LR(∅)) NECESSITO DI QUALCOSA DI PIÙ POTENTE

PILA	INPUT	OPERAZIONE
$I_\emptyset$	a a b b ↙	SHIFT
I_0, a, I_3	a b b ↙	SHIFT
I_0, a, I_3, a, I_3	b b ↙	SHIFT
$I_0, a, I_3, \boxed{a, I_3, b, I_6}$	b ↙	REDUCE $A \to ab$
$I_0, a, I_3, \boxed{A, I_2}$	b ↙	REDUCE $S \to A$
I_0, a, I_3, S, I_7	b ↙	SHIFT
$I_0, a, I_3, S, I_7, b, I_8$	↙	REDUCE $A \to aSb$
$I_0, \boxed{A, I_2}$	↙	REDUCE $S \to A$
I_0, S, I_2	↙	SHIFT
$I_0, \boxed{S, I_2, ↙, I_4}$		REDUCE $S' \to S↙$
I_0, S		ACCEPT

ALBERO DI DERIVAZIONE