1. L(DFA) = L(NFA) = REG (Pag 12) ✓
2. L(re)=L(DFA)=REG (Pag 16) ✓
3. Pumpling Lemma (Pag 20) ✓
4. Per ogni CFG, esiste una CFG equivalente (che genera lo stesso linguaggio) in forma normale (Pag 23) ✗
5. Un linguaggio è acontestuale (generato da una CFG) se e solo se esiste un PDA che lo riconosce (pag 27) ✗
6. Pumpling Lemma CFG (Pag 32)

1) $L(DFA) \subseteq L(NFA)$

È OVVIO PERCHÉ UN NFA È UN DFA CHE PRESENTA ANCHE ε-ARCHI

2) $L(NFA) \subseteq L(DFA)$

DIMOSTREREMO PER COSTRUZIONE

COSTRUIREMO UN DFA D PARTENDO DA UN NFA N

- $Q_D = P(Q_N)$
- $F_D = \{R \in Q_D \mid R \cap F_N = \phi\}$
- DATO $R \in Q_D$

$$E(R) = \{q \in Q_N \mid \exists p \in R \mid p \text{ raggiunge } q \text{ con } \varepsilon\text{-archi}\}$$

- $q_{0_D} = E(q_{0_N})$
- DATI $R \in Q_D$ e $a \in \Sigma$

$$\delta_D(R,a) = \bigcup_{z \in R} E(\delta_N(z,a))$$

È CHIARO PER COSTRUZIONE CHE $w \in L(N) \Leftrightarrow w \in L(D)$

□

2) $L(re) = L(NFA) = REG$

UNA RE È UNA STRINGA DA RAPPRESENTARE UN LINGUAGGIO O VICEVERSA CHE

$\phi \in re(\Sigma)$

$\varepsilon \in re(\Sigma)$

$a \in re(\Sigma)$ dove $a \in \Sigma$

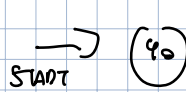
$R_1, R_2 \in re(\Sigma)$ $\quad R_1 \cup R_2 \in re(\Sigma)$

$R_1, R_2 \in re(\Sigma)$ $\quad R_1 \circ R_2 \in re(\Sigma)$

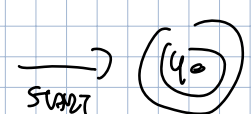
$R_1, R_2 \in re(\Sigma)$ $\quad R_1^* \in re(\Sigma)$

1° $L(re) \subset L(NFA)$

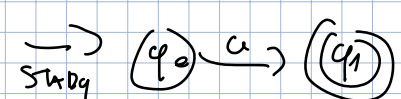
È CHIARO CHE POSSIAMO SIMULARE UN NFA CHE ACCETTA RE



$= \phi \in re(\Sigma)$



$= \varepsilon \in re(\Sigma)$



$= a \in re(\Sigma)$

C'È ANCHE QUINDI POSSIBILE COSTRUIRE UN NFA CHE ACCETTA QUALSIASI CW ACCETTATA DA RE

2° PER DIMOSTRARE $L(NFA) \subset L(GE)$ USEREMO I GNFA

CHE HANNO:

- 1 STATO ACCETTANTE
- TUTTI ARCHI ESCENO da q START
- TUTTI ARCHI ENTRANO in q ACC
- per ogni coppia di STATI c'è un ARCO
- gli ARCHI HANNO come ETICHETTE ESPRESSIONI REGOLARI

È CHIARO CHE $L(DFA) = L(NFA) \subset L(GNFA)$

POICHÉ PARTENDO DA UN DFA / NFA POSSO CREARE UN GNFA EQUIVALENTE.

PER DIMOSTRARE $L(GE) \subseteq L(GNFA)$ USEREMO LA FUNZIONE

$Converti(G) = G'$ (UN GNFA con UNO STATO in MENO EQUIVALENTE)

IL RESTO DELLA DIMOSTRAZIONE È FATTA A MENTE PER VOGLIA DI SCRIVERE

3) PUMPING LEMMA

SIA $L \in REG$ E SIA $w \in L$ $\exists\ p \leq |w|$

DETTO PUMPING LC È POSSIBILE SCOMPORRE $w = xyz$

CON $|y| \neq 0$ $|xy| \leq p$ E $xy^i z \in L$

DIM: SIA $p = |Q|$ DI UN DFA CHE ACCETTA L

SIA $w = w_1, w_2, \dots w_n$ E SIA $R = \{r_1 \dots r_m\}$ GLI STATI IN SUCCESSIONE TC $\forall i\ (r_i, w_i) = r_{i+1}$

ORA p QUINDI $m+1 > p \Rightarrow$ IN R C'È ALMENO UNO STATO RIPETUTO CHIAMIAMO R_i LA PRIMA RIPETIZIONE E R_j L'ULTIMO

DUNQUE w CAMBIA SE QUE:

$x = w_1 \dots w_{i-1}$

$y = w_i \dots w_{j-1} \Rightarrow |xy| \leq p$ E $j \neq i \Rightarrow |y| \neq 0$

$z = w_j \dots w_n$

INDUCE FACILE $\forall$ E L PORTANDO y IN SE STESSO $xy^i z \in L$

□

ESEMPIO DI APPLICAZIONE

$w \in L = \{0^n 1^n \mid n \in \mathbb{N}\}$

SIA $p = n \Rightarrow w = xyz$ USIAMO UNA DELLE TANTE SCOMPOSIZIONI

$w = 0^n 1^{n-q} 1^q \Rightarrow 0^{n+(n-q)j} 1^q \notin L$

4) PER OGNI CFG ESISTE UNA CFG IN FORMA NORMALE EQUIVALENTE

UNA CFG IN FORMA NORMALE RISPETTA ALCUNE REGOLE

- $A \to BC$ con $B, C \in V$
- $A \to a$ $a \in \Sigma$ $\in B \neq S \in C \neq S$
- $S \to \varepsilon$ PUO' ESSERE

LA DIMOSTRAZIONE DI UNA PROCEDURA DI SEGUITO RIPORTATA

SIA

$S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \varepsilon$

1° AGGIUNGO S_0 CHE INDICA ε LA REGOLA $S_0 \to S$

$S_0 \to S$

$S \to ASA \mid aB$

$A \to B \mid S$

$B \to b \mid \varepsilon$

2° ELIMINO LE ε-REGOLE $(B \to \varepsilon)$

1) $S_0 \to S$

$S \to ASA \mid aB \mid a$

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

$B \to b$

2) $A \to \varepsilon$

$S_0 \to S$

$S \to ASA \mid aB \mid a \mid SA \mid AS$

$A \to B \mid S$

$B \to b$

3° ELIMINO TUTTE LE REGOLE UNITARIE ($S_0 \to S$, $A \to B$, $A \to S$)

$S_0 \to ASA \mid aB \mid a \mid SA \mid AS \mid S$

$A \to b \mid ASA \mid aB \mid a \mid SA \mid AS \mid S$

$B \to b$

4° SEPARO LE REGOLE CON 3 O PIÙ ELEMENTI ($S_0 \to ASA$ e $A \to ASA$)

$S_0 \to AA_1 \mid aB \mid a \mid SA \mid AS \mid S$

$A \to b \mid AA_1 \mid aB \mid a \mid SA \mid AS \mid S$

$B \to b$

$A_1 \to SA$

5° TOLGO LE REGOLE CON 2 ELEMENTI DI CUI UNO TERMINALE ($S_0 \to aB$, $A \to aB$)

$S_0 \to AA_1 \mid UB \mid a \mid SA \mid AS \mid S$

$A \to b \mid AA_1 \mid UB \mid a \mid SA \mid AS \mid S$

$B \to b$

$A_1 \to SA$

$U \to a$ □

⇒ UN LINGUAGGIO È GENERATO DA UNA CFG ⇔
⇒ DA UN PDA P CHE LO RICONOSCE

⇒)

NOTA Bene: USEREMO UNA NOTAZIONE RISTRETTA DEL PDA IN CUI INVECE DI INSERIRE E LEGGERE UN CARATTERE ALLA VOLTA USIAMO DIRETTAMENTE DELLE STRINGHE, USEREMO INOLTRE 3 STATI q_{START}, q_{LOOP}, q_{ACCEPT}.

L'IDEA È QUELLA DI SIMULARE UNA CFG CON UN PDA SFRUTTANDO IL NON DETERMINISMO. L'ALGORITMO DA USARE È IL SEGUENTE

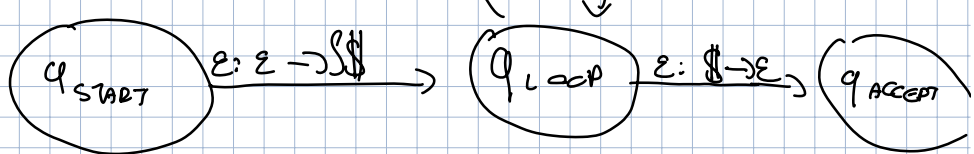
WHILE TRUE:
- SE NEL TOP DELLO STACK C'È UNA VARIABILE ESEGUE TUTTE LE REGOLE DI DERIVAZIONE SFRUTTANDO IL NON DET
- SE C'È UN TERMINALE LO RIMUOVE E CONTINUA CHE IL PROSSIMO CARATTERE SIA LO STESSO SE SI CONTINUA ALTRIMENTI RIFIUTA
- SE NEL TOP C'È $ (STACK VUOTO) SE HA LETTO TUTTA LA STRINGA DI INPUT ACCETTA ALTRIMENTI RIFIUTA

ESEMPIO:

$S \to aTb \mid b$
$T \to Ta \mid \varepsilon$

$\varepsilon: S \to aTb$
$\varepsilon: T \to Ta$
$b: b \to \varepsilon$
$a: a \to \varepsilon$
$\varepsilon: T \to \varepsilon$
$\varepsilon: S \to b$

IL PDA È



$\Leftarrow$) L'IDEA È QUELLA DI CONSIDERARE PER OGNI COPPIA DI STATI p, q C'È UNA VARIABILE A_{pq} CHE GENERA LE STRINGHE CHE PERMETTONO AL PDA DI PASSARE DA p A q CON LA PILA VUOTA

IL RESTO DELLA DIMOSTRAZIONE NON SCRITTO