Revisão MDI:

- **Conjunto:** coleção de elementos em que não há ocorrências múltiplas do mesmo elemento.
- **Símbolo:** elemento de representação gráfica unica e indivisível
- **Número de elementos:** indica o tamanho do conjunto. Notação: $|A|$
- **Pertencimento:** indica se um elemento está no conjunto. Ex: $a \in A$ ou $b \notin A$.
- **Conjunto finito:** possui número finito de elementos.
- **Conjunto infinito:** possui número infinito de elementos.
- **Conjunto vazio:** conjunto sem elementos. Notação $\{\}$ ou $\emptyset$.
- **Subconjunto:** A está contido em B se todo elemento de A está em B, dizemos que A é subconjunto de B. Notação: $A \subseteq B$
- **União:** Junta elementos de dois conjuntos. Notação: $A \cup B$
- **Interseção:** elementos comuns entre dois conjuntos. Notação: $A \cap B$
- **Conjuntos disjuntos:** não tem elementos em comum: $A \cap B = \emptyset$
- **Diferença:** $A - B = \{x \mid x \in A \text{ e } x \notin B\}$
- **Complementação:** $\bar{A} = U - A = \{x \mid x \in U \text{ e } x \notin A\}$
- **Produto Cartesiano:** Pares ordenados formados por elementos de dois conjuntos. Notação: $A \times B = \{(a,b) \mid a \in A \text{ e } b \in B\}$

Elementos básicos LFA:

- **Alfabeto:** conjunto finito de símbolos. Ex: $\Sigma = \{a, b\}$
- **Palavra:** sequência finita de símbolos de um alfabeto.
  - ε: palavra vazia, sem símbolos.
- **Tamanho da palavra:** número de símbolos de uma palavra. Notação: $|w|$
  - $|w|_a$: número de símbolos "a" na palavra w.
- **Conjunto de palavras sobre Σ:**
  - Σ^*: conjunto de todas as palavras com símbolos de Σ.
  - Σ^+: conjunto de todas as palavras não vazias com símbolos de Σ.
    - $\Sigma^+ = \Sigma^* - \{\varepsilon\}$
- **Prefixo:** qualquer início de uma palavra. Ex: ab é prefixo de "abba"
- **Sufixo:** qualquer final de uma palavra. Ex: ba é sufixo de "abba"
- **Subpalavra:** sequência contínua dentro de uma palavra. Ex: bb é subpalavra de "abba"
  - Prefixo e sufixo são subpalavras
- **Linguagem formal:** conjunto de palavras sobre um alfabeto. Qualquer subconjunto de Σ^*.
- **Concatenação:** junção de duas palavras. Ex: "ab" . "ba" = "abba"
  - **Propriedades:**
    - *Associatividade: $v(wt) = (vw)t$
    - *Elemento Neutro (esq/dir): $\varepsilon w = w = w\varepsilon$
- **Concatenação Sucessiva:** Repetição de uma palavra. Notação: w^n
  Ex: w = "ab", w^3 = "ababab"
- **Gramática:** permite gerar todas as palavras da linguagem que representa.

$$G = (V, T, P, S)$$

- V: conjunto finito de variáveis (ou símbolos não terminais), usados para gerar palavras.
- T: conjunto finito de símbolos terminais, disjuntos de V. Símbolos que aparecem na palavra final.
- P: conjunto de produções, ou regra de derivação, define como as palavras podem ser substituídas
- S: símbolo inicial, $S \in V$.

- **Derivação:** é um processo de aplicar regras da gramática para transformar um símbolo inicial em uma palavra feita só de terminais
  - $\Rightarrow$: representa um passo de derivação
  - $\Rightarrow^*$: representa zero ou mais passos de derivação
  - $\Rightarrow^+$: representa um ou mais passos de derivação
  - $\Rightarrow^i$: representa i passos de derivação
- **Linguagem Gerada:** linguagem gerada por Gr. $L(G_1)$ ou $Gera(G_1)$
  - $L(G_1) = \{w \in T^* \mid S \Rightarrow^+ w\}$
- **Equivalência de Gramáticas:** G_1 e G_2 são equivalentes se: $Gera(G_1) = Gera(G_2)$

Linguagens Regulares:

são linguagens que podem ser descritas de forma simples e mecânicas por:

- Autômatos Finitos (AFD, AFN e AFε) → Reconhecedor
- Gramática regular → Gerador
- Expressão regular → Denotacional

Autômato Finito Determinístico (AFD)

- **Máquina composta por:**
  - **Fita:** contém a informação a ser processada (símbolos)
  - **Unidade de controle:** armazena o estado atual do autômato, faz a leitura da fita, movimento-se para a direita.
  - **Programa:** define o comportamento da máquina.
- **5-upla:** $M = (\Sigma, Q, \delta, q_0, F)$
  - Σ: alfabeto
  - Q: conjunto de todos os estados
  - δ: função programa → $\delta: Q \times \Sigma \to Q$
  - q_0: estado inicial → $q_0 \in Q$
  - F: conjunto de estados finais → $F \subseteq Q$
- **Função Programa estendida**
  - $\hat{\delta}: Q \times \Sigma^* \to Q$: define o estado após ler toda a palavra w tal que $w \in \Sigma^*$
    - Ex: $\hat{\delta}(q, w)$ = estado resultante após ler toda palavra.
    - **Propriedades:**
      - * $\hat{\delta}(q, \varepsilon) = q$
      - * $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ tal que $a \in \Sigma$ e $w \in \Sigma^*$
- **Um AFD sempre pára:** pois qq palavra de entrada é finita
- **Para no fim da fita:**
  - aceita: se atinge um estado final
  - rejeita: se atinge um estado não-final
- **Linguagem aceita por um AFD:** $L(M)$ ou $ACEITA(M)$
- **Linguagem rejeitada por um AFD:** $REJEITA(M)$

Autômato Finito Não-determinístico (AFN)

- Qualquer AFN pode ser simulado por um AFD.
- $\delta: Q \times \Sigma \to 2^Q$: ou seja δ recebe um par (estado, símbolo) e retorna um subconjunto de Q.
  - Se em alguma linha da função programa tivermos um subconjunto de Q, esse autômato é um AFN.
- **Função Programa Estendida:** $\hat{\delta}: 2^Q \times \Sigma^* \to 2^Q$
  - Seja $M = (\Sigma, Q, \delta, q_0, F)$ um AFN e $P \in 2^Q$:
    - * $\hat{\delta}(P, \varepsilon) = P$
    - * $\hat{\delta}(P, aw) = \hat{\delta}(\bigcup_{q \in P} \delta(q, a), w)$ (união de todos os conjuntos de estados de q que podem ser acessados a partir de cada estado de P lendo a)

Algoritmo AFN para AFD

- Entrada: $AFN(Q, \Sigma, \delta, q_0, F)$
  Saída: $AFD(Q', \Sigma, \delta', q_0', F')$
- 1º: o estado inicial do AFD é $\{q_0\}$
- 2º: enquanto houver estados não processados:
  - ↳ para cada símbolo $a \in \Sigma$, calcule:
    - * a união de $\delta(q,a)$ para todo q no conjunto atual
    - * se esse novo conjunto não estiver na lista adicione.
- 3º: marque como finais os conjuntos que contêm algum estado final do AFN.
- Ex:

AFN

$q_0 = q_0$
$F = \{q_2\}$

δ	a
q_0	$\{q_1, q_2\}$
q_1	$\{q_2\}$
q_2	$\emptyset$

→ AFD

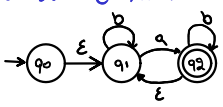
$q_0 = \{q_0\}$
$F = \{\{q_1,q_2\}, \{q_2\}\}$

δ	a
$\{q_0\}$	$\{q_1, q_2\}$
$\{q_1,q_2\}$	$\{q_2\}$
$\{q_2\}$	$\emptyset$

Autômato Finito com Movimentos Vazio (AFNε ou AFε)

- Transições sem leitura de símbolo da fita.
- Qualquer AFN com movimentos vazio pode ser simulado por um AFN
- Fecho-ε ou F_ε: seja $M = (\Sigma, Q, \delta, q_0, F)$ um AFε
  - ↳ $F_\varepsilon : Q \to 2^Q$
    - ↳ se $\delta(q,\varepsilon)$ não é definido então $F_\varepsilon(q) = \{q\}$
    - ↳ se $\delta(q,\varepsilon)$ é definido $F_\varepsilon(q) = \{q\} \cup \delta(q,\varepsilon) \cup_{p \in \underline{\delta}(q,\varepsilon)} F_\varepsilon(p)$
      - ↳ Conjunto de todos os estados que podem ser acessados com zero ou mais transições vazias (ε).
- Para conjunto de estados
  - ↳ $F_\varepsilon : 2^Q \to 2^Q$ tq $F_\varepsilon(P) = \cup_{q \in P} F_\varepsilon(q)$, e $P \in 2^Q$
    - ↳ União de $\{F_\varepsilon(q) \mid q \in P\}$

- Função Programa Estendido: $\underline{\delta} : 2^Q \times \Sigma^* \to 2^Q$
  - ↳ $\underline{\delta}(P, \varepsilon) = F_\varepsilon(P)$ tq $P \in 2^Q$
  - ↳ $\underline{\delta}(P, wa) = F_\varepsilon(R)^*$ ← após achar R faz $F_\varepsilon(R)^*$
    - ↳ $R = \{r \mid r \in \delta(s,a)$ e $s \in \underline{\delta}(P,w)\}$
      - ↳ onde: $a \in \Sigma$, $w \in \Sigma^*$
- Algoritmo AFε → AFN
  - ↳ 1º → Calcular $F_\varepsilon(q)$ $\forall q \in Q$, todos os estados alcançáveis a partir de q usando apenas transições ε.
  - ↳ 2º → para cada $q \in Q$ e cada $a \in \Sigma$:
    - ↳ (I) Inicialize $R = \emptyset$
    - ↳ (II) Para cada estado $p \in F_\varepsilon(q)$ adicione $\delta(p,a)$ em R (estados alcançados com a)
    - ↳ (III) Faça:
      - * $\delta'(q,a) = \cup F_\varepsilon(r)$ $\forall r \in R$, ou seja F_ε de todos os estados alcançado com a.
  - ↳ 3º → Novos estados finais: um estado $q \in Q$ será final em F' se $(F_\varepsilon(q) \cap F) \neq \emptyset$, ou seja se o $F_\varepsilon(q)$ contém algum estado final original, então q também é final.
- Ex: $M = \{\{q_0,q_1,q_2\}, \{a,b\}, q_0, \{q_2\}\}$

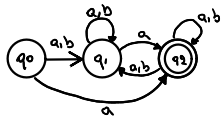


- $F_\varepsilon(q)$ $\forall q \in Q$
  $F_\varepsilon(q_0) = \{q_0, q_1\}$
  $F_\varepsilon(q_1) = \{q_1\}$
  $F_\varepsilon(q_2) = \{q_1, q_2\}$
- Construir δ'

$\delta'(q_0,a) = F_\varepsilon(\{r \mid r \in \delta(s,a)$ e $s \in \underbrace{\underline{\delta}(q_0,\varepsilon)}_{= F_\varepsilon(q_0)}\}) = F_\varepsilon(\{q_2\}) = \{q_1,q_2\}$
$\delta'(q_0,b) = F_\varepsilon(\{r \mid r \in \delta(s,b)$ e $s \in \underline{\delta}(q_0,\varepsilon)\}) = F_\varepsilon(\{q_1\}) = \{q_1\}$
$\delta'(q_1,a) = F_\varepsilon(\{r \mid r \in \delta(s,a)$ e $s \in \underline{\delta}(q_1,\varepsilon)\}) = F_\varepsilon(\{q_2\}) = \{q_1,q_2\}$
$\delta'(q_1,b) = F_\varepsilon(\{r \mid r \in \delta(s,b)$ e $s \in \underline{\delta}(q_1,\varepsilon)\}) = F_\varepsilon(\{q_1\}) = \{q_1\}$
$\delta'(q_2,a) = F_\varepsilon(\{r \mid r \in \delta(s,a)$ e $s \in \underline{\delta}(q_2,\varepsilon)\}) = F_\varepsilon(\{q_2\}) = \{q_1,q_2\}$
$\delta'(q_2,b) = F_\varepsilon(\{r \mid r \in \delta(s,b)$ e $s \in \underline{\delta}(q_2,\varepsilon)\}) = F_\varepsilon(\{q_1,q_2\}) = \{q_1,q_2\}$

- logo temos:

δ	a	b
q_0	$\{q_1,q_2\}$	$\{q_1\}$
q_1	$\{q_1,q_2\}$	$\{q_1\}$
q_2	$\{q_1,q_2\}$	$\{q_1,q_2\}$

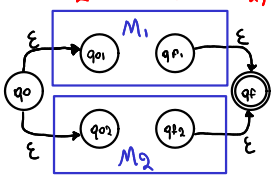


Expressões Regulares: formalismo denotacional

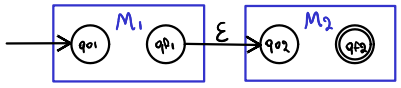
- ∅ é ER, e denota a linguagem vazia
- ε é ER, e denota a linguagem $\{\varepsilon\}$
- x é ER, onde $x \in \Sigma$ e denota a linguagem $\{x\}$
- Se r e s são ER, e denotam as linguagens R e S respectivamente, então:
  - ↳ (r+s) é ER, e denota R ∪ S
  - ↳ (rs) é ER, e denota RS
  - ↳ (r^*) é ER, e denota R^*
- Precedência de parênteses em ER:
  - ↳ 1º: Concatenação sucessivo (* ou +)
  - ↳ 2º: Concatenação
  - ↳ 3º: União
- Exemplos:

aa - somente a palavra aa
ba^* - todas as palavras que iniciam por b, seguido por zero ou mais a's
$(a+b)^*$ - todas as palavras sobre $\{a,b\}$
$(a+b)^* aa (a+b)^*$ - todas as palavras com aa como subpalavra
$a^* ba^* ba^*$ - todas palavras contendo exatamente dois b's
$(a+b)^* (aa+bb)$ - todas palavras que terminam com aa ou bb
$(a+\varepsilon)(b+ba)^*$ - todas palavras que não possuem dois a's consecutivos.

- Suponde: $\begin{cases} M_1 = \{\Sigma_1, Q_1, \delta_1, q_{01}, \{q_{f1}\}\} \text{ tq } L(M_1) = \text{Gera}(r_1) \\ M_2 = \{\Sigma_2, Q_2, \delta_2, q_{02}, \{q_{f2}\}\} \text{ tq } L(M_2) = \text{Gera}(r_2) \end{cases}$
  - ↳ $r = r_1 + r_2 \to M = \{\Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2 \cup \{q_0, q_f\}, \delta, q_0, \{q_f\}\}$



  - ↳ $r = r_1 r_2 \to M = \{\Sigma_1 \cup \Sigma_2, Q_1 \cup Q_2, \delta, q_{01}, \{q_{f2}\}\}$



  - ↳ $r = r_1^* \to M = \{\Sigma_1, Q_1, \delta, q_0, \{q_f\}\}$