

Dezembro
de 2021



Identificação

Universidade de Pernambuco (UPE)

Escola Politécnica de Pernambuco (Poli)

Semestre: 2021.1

Disciplina: Matemática Discreta / Programação II

Docente: Dr. Ruben Carlo Benante

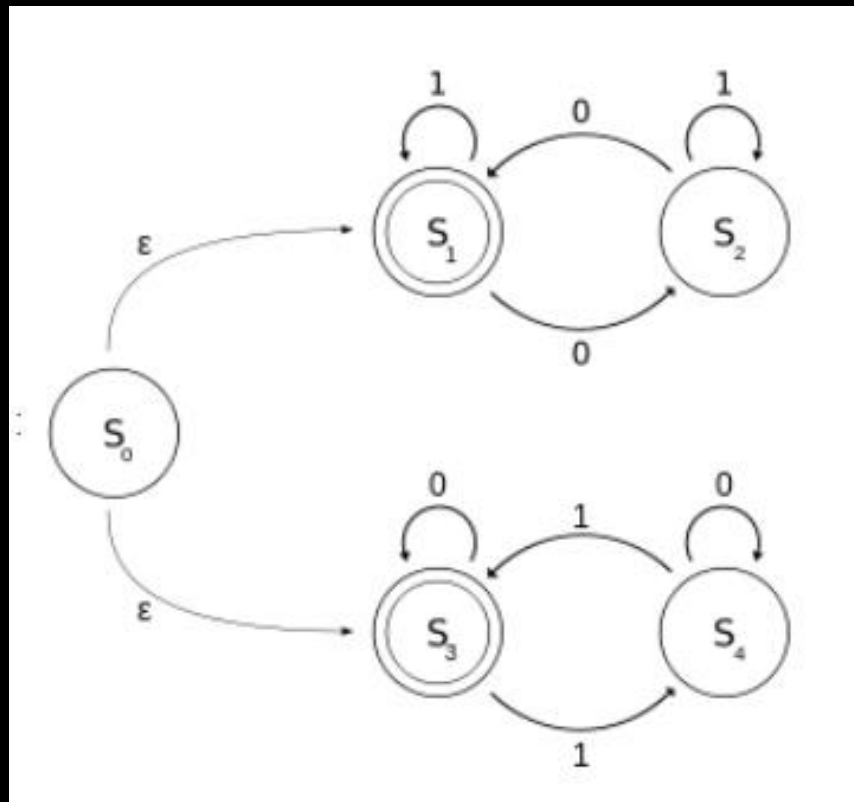
Grupo 02 - Xavier

- Guilherme Rodrigues Chaves do Nascimento
- João Alves Pereira Neto
- **Thiago de Azevedo Cavendish**
- Ulisses Mosart Sobrinho
- Maria Isabel do Nascimento Freitas

Teoria dos Autômatos

❖ Introdução

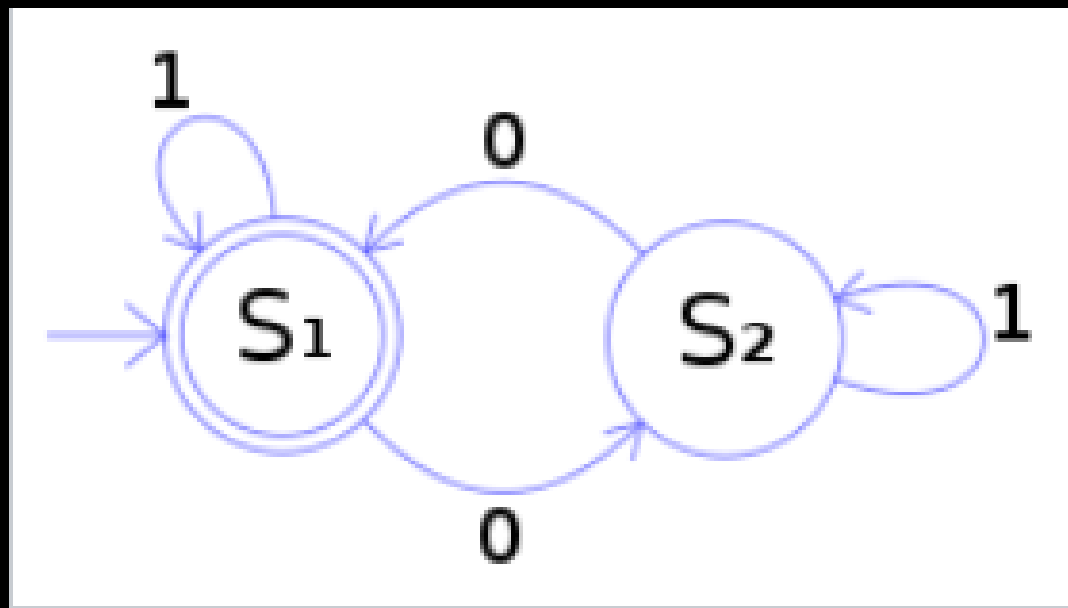
- **AFND** é uma máquina de estados finita onde para cada par de estado e símbolo de entrada pode haver **vários** próximos estados possíveis.



Teoria dos Autômatos

❖ Introdução

- **AFD** é uma máquina de estados finitos onde, dada uma configuração e um símbolo da cadeia de entrada, existe **somente um** estado para o qual a máquina pode transitar.



Teoria dos Autômatos

❖ Introdução

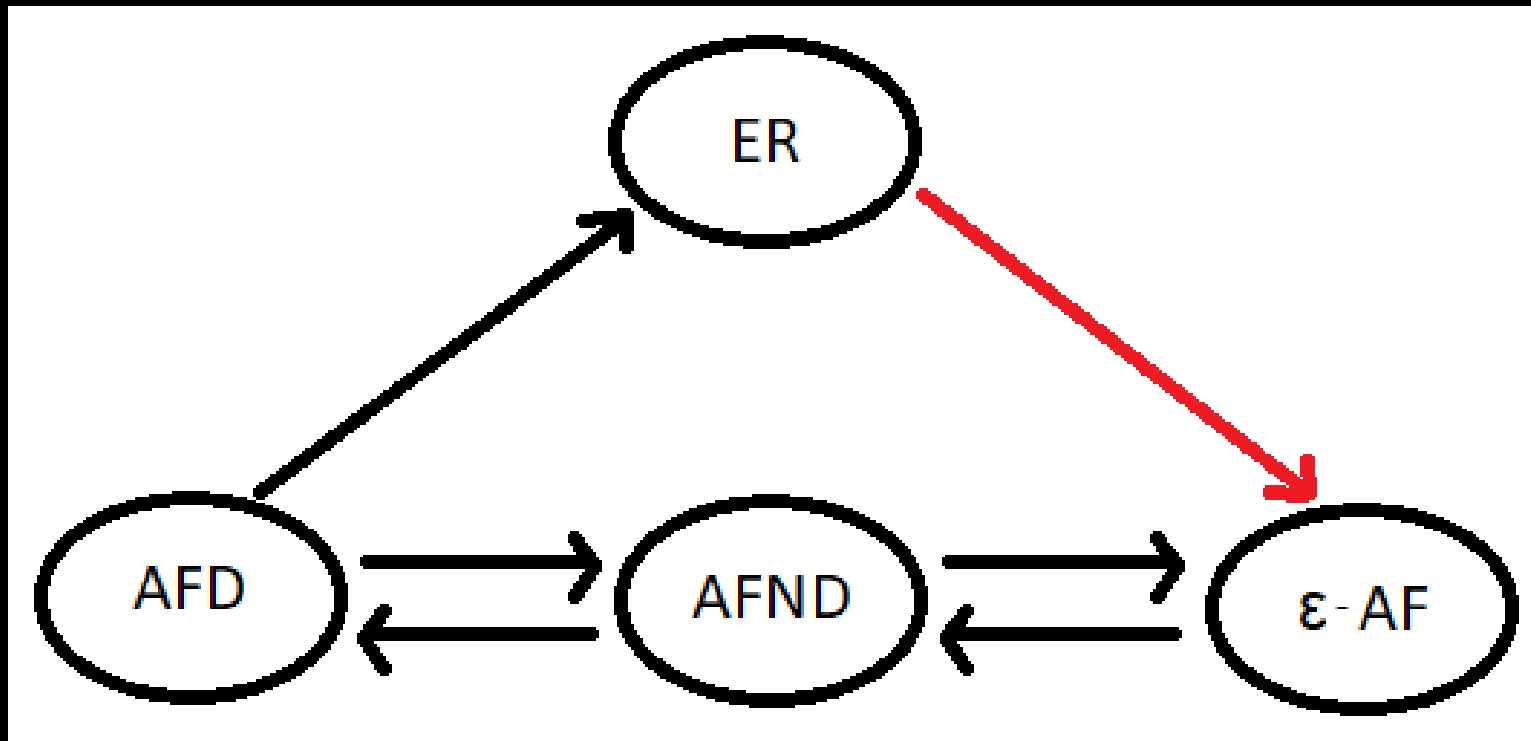
- ER provê uma forma concisa e flexível de identificar **cadeias de caracteres** de interesse, como caracteres particulares, palavras ou padrões de caracteres.
- É escrita numa linguagem formal que pode ser interpretada por um processador de expressão regular.

$$1^*(0(1^*)0(1^*))^*$$

Teoria dos Autômatos

❖ Conversão

- AF e ER representam exatamente o mesmo conjunto de linguagens, as **Linguagens Regulares**.



Conversor de Autômatos

❖ Entrada de dados - tipos

- Uma quintupla que pode representar um AFD ou um AFND
- Uma string que representa uma expressão regular.

❖ Entrada de dados - fontes

- Leitura de arquivo
- Leitura da entrada padrão (manual)

Conversor de Autômatos

❖ Flags

- -n = a entrada de dados é uma quintupla AFND que será convertida para AFD
- -d = a entrada de dados é uma quintupla AFD que será convertida para uma ER
- -e = a entrada de dados é uma string ER que será convertida em um AFND
- -f = faz a entrada de um arquivo
- -h = mostra o help e termina
- -c = mostra a versão e o copyright e termina

Conversor de Autômatos

❖ Formato da quintupla (AF)

- #K = número de estados
- #A = final do alfabeto
- #S = estado inicial (único)
- #F = lista de estados finais
- #D = função delta (transições)

Conversor de Autômatos

❖ Formato da string (ER)

- a-z = símbolos do alfabeto
- ξ = símbolo vazio (epsilon ou “E”)
- | = disjunção (ou)
- . = conjunção (e)
- * = repete o símbolo anterior zero ou mais vezes (estrela de Kleene)
- () = muda a prioridade de execução das operações

Conversor de Autômatos

❖ Processamento

- Determinar o tipo de conversão a ser realizado (pelas flags)
- Receber a entrada (arquivo ou manual)
- Imprimir a resposta

Conversor de Autômatos

❖ Entrada e leitura dos arquivos

- `char name_file[SBUFF];`
- `char str[SBUFF];`
- `FILE *arq = NULL;`
- `t_quintupla`

Conversor de Autômatos

❖ Entrada e leitura dos arquivos

- `while`((opt = getopt(argc, argv, "vhcf:nde")) != EOF)
 `switch`(opt)
 {

• `case 'f':`
 `strncpy`(name_file, optarg, SBUFF);
 `entrada` = 0;
 `break`;

Conversor de Autômatos

❖ Entrada e leitura dos arquivos

- `case 'n':`
 `convert = 1;`
 `break;`
- `case 'e':`
 `convert = 3;`
 `break;`

Conversor de Autômatos

❖ Entrada e leitura dos arquivos

- `if(entrada)`
 {
 `if(convert == 1 || convert == 2)`
 `m_recebe_AFND(&AFND);`
 `else`
 {
 `arq = fopen("temp.txt", "a");`
 `scanf("%s", str);`
 `fprintf(arq, "%s", str);`
 `fflush(arq);`
 `read = fopen("temp.txt", "r");`
 `le_arquivo(read);`
 `remove("temp.txt");`
 }
 }
 }

Conversor de Autômatos

❖ Entrada e leitura dos arquivos

- `else`
 {
 file = fopen(name_file, "r");

 if(!file)
 exit(EXIT_SUCCESS);
 }

Conversor de Autômatos

❖ Conversão

- `switch(convert)`
`{`
- `case 1:` `// AFND -> AFD //`
 `if(entrada == 0)`
 `recebe_AFND(file, &AFND);`
 `AFD = Calc_AFD(AFND);`
 `print_quintupla(AFD);`
 `break;`

Conversor de Autômatos

❖ Conversão

- **case 3:**

```
if(entrada == 0)  
    le_arquivo(file);  
break;
```

// ER -> AFND //

Conversor de Autômatos

❖ Outras funções

- `t_lef *Calc_E_f(t_quintupla *q, unsigned short int num, char *test);`
- `t_lef *Calc_novos_finais(t_quintupla *q, t_list_lef *base);`
- `t_lef *Calc_E(t_quintupla *q, t_lef *ei, char *test);`
- `t_lef *Calc_epsilon(t_quintupla *q, unsigned short int num);`
- `t_lef *Calc_e_i(t_quintupla *q, unsigned short int num);`
- `int ord_novo(t_list_lef *list, t_lef *lef);`
- `int buscar(t_lef *est, t_lef *est2);`
- `void juntar_lef_list(t_list_lef *est, t_list_lef **est2);`

Conversor de Autômatos

❖ Outras funções

- `void le_arquivo(FILE *exp_reg);`
- `void divide_expressao(t_delta conteudo1);`
- `void conversao_er_AFND(char linha[SBUFF], t_quintupla *novo_AFND, unsigned short int *est);`
- `void organiza_listas(t_delta **cabeca, unsigned short int i, char a, unsigned short int f);`

Conclusão

#K	#K
4	4
#A	#A
C	c
#S	#S
3	0
#F	#F
1 3	0 2 3
#D	#D
0 a 0	0 a 1
0 b 1	0 c 0
0 E 2	1 a 2
1 a 1	1 b 3
1 b 1	1 c 0
1 E 2	2 a 2
2 a 1	2 b 3
2 b 2	2 c 0
2 c 3	3 a 3
3 a 0	3 b 3
3 c 3	3 c 0



❖ Conversão AFND -> AFD

Dada a entrada teste (quíntupla AFND), podemos executar o programa automax usando o seguinte comando com pipe:

```
cat 02-teste-AFND.txt | ./automax.x -n
```