

Aspectos Teóricos da Computação

Prof. Rodrigo Martins

`rodrigo.Martins@francomontoro.com.br`

Cronograma

- ◆ Autômatos Finitos Não Determinísticos (AFND)
- ◆ Exemplos
- ◆ Exercícios

Autômatos Finitos

- ◆ Autômato Finito Determinístico (AFD)
 - É um sistema de estados finitos onde para cada símbolo do alfabeto existe somente uma saída de um estado n .
- ◆ Autômato Finito Não Determinístico (AFND)
 - De um determinado estado podem sair duas ou mais transições com o mesmo símbolo para estados diferentes.

Autômato Finito Não Determinístico (AFND)

- ◆ Um AFND tem o poder de estar em vários estados ao mesmo tempo.
- ◆ Habilidade de, ao ler uma entrada estando em um determinado estado, poder escolher entre várias possibilidades, o próximo estado. Há mais que uma opção para o próximo estado.
- ◆ Trabalha com possibilidades.
- ◆ Em algumas situações podemos converter AFND em AFD.

Diferença entre AFD e AFND

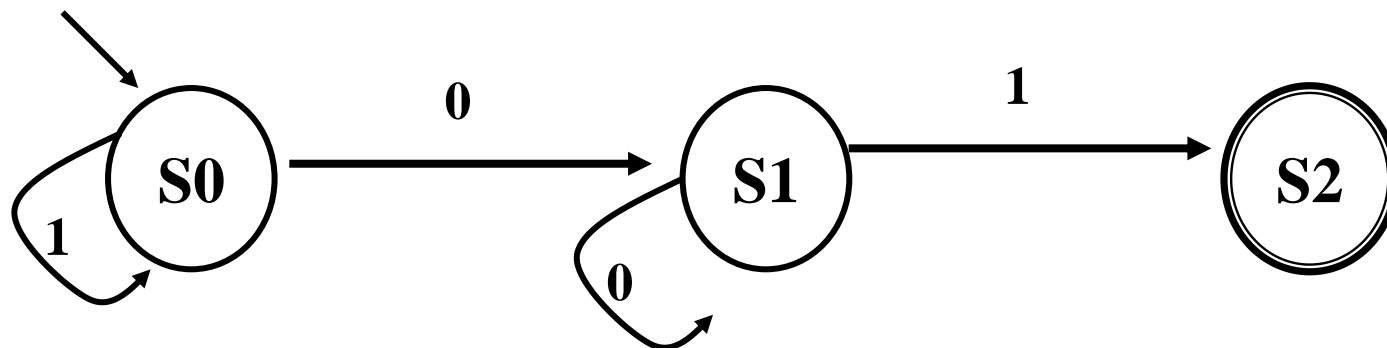
- ◆ **AFD:**

- Em um AFD, para cada estado e símbolo de entrada, existe exatamente uma transição de estado, ou seja, dada uma entrada específica, o AFD só tem um caminho possível a seguir.

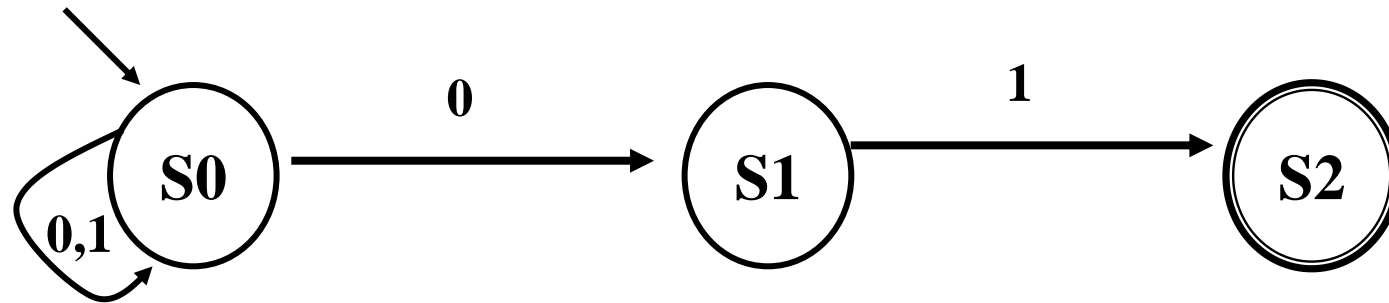
- ◆ **AFND:**

- Em um AFND, para um estado e um símbolo de entrada, pode haver várias transições possíveis, incluindo nenhuma ou várias para diferentes estados.
- Sendo assim, o AFND pode escolher entre múltiplas opções (ou nenhuma) para uma entrada dada.

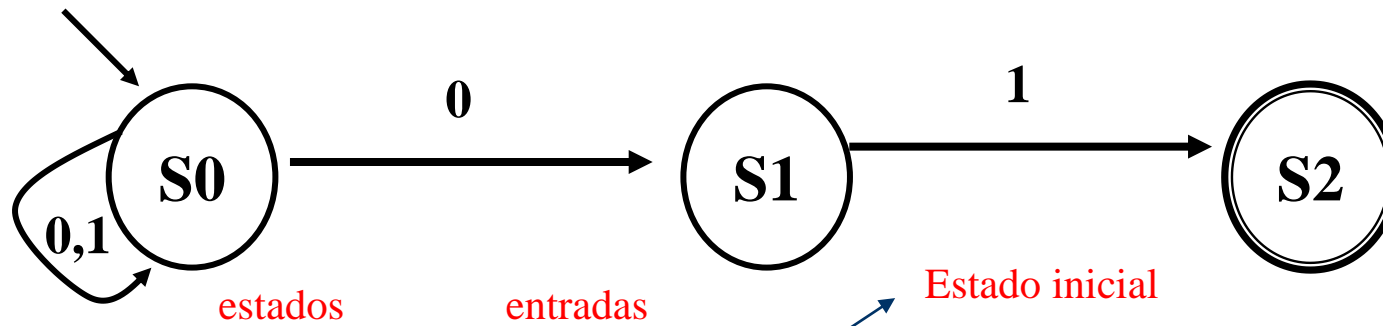
AFD – Autômato Finito Determinístico



AFND - Autômato Finito NÃO Determinístico



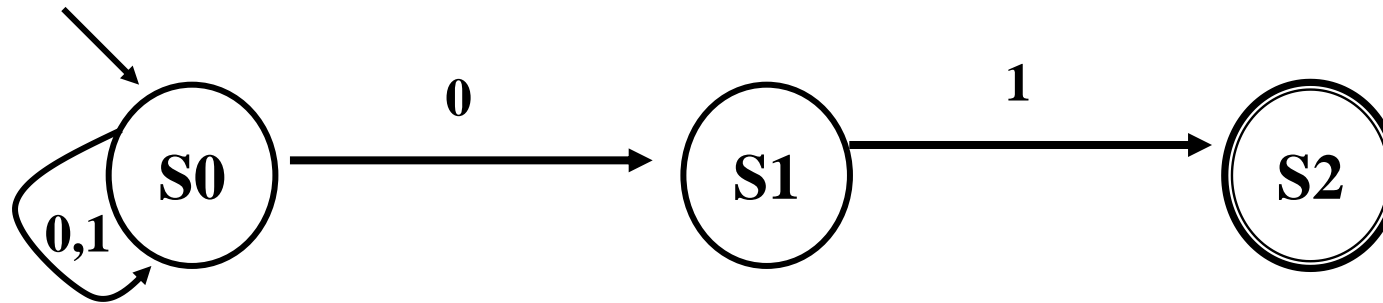
Formalizando AFND



- ♦ $M = (\{s0, s1, s2\}, \{0, 1\}, \mathbf{T}, s0, \{S2\})$.
- ♦ **T é a tabela de transição:**

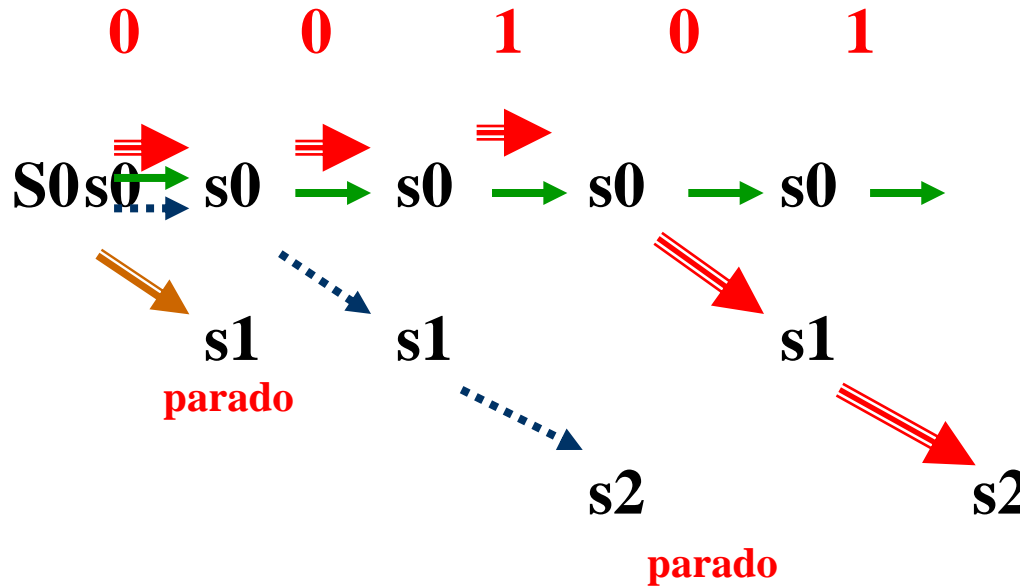
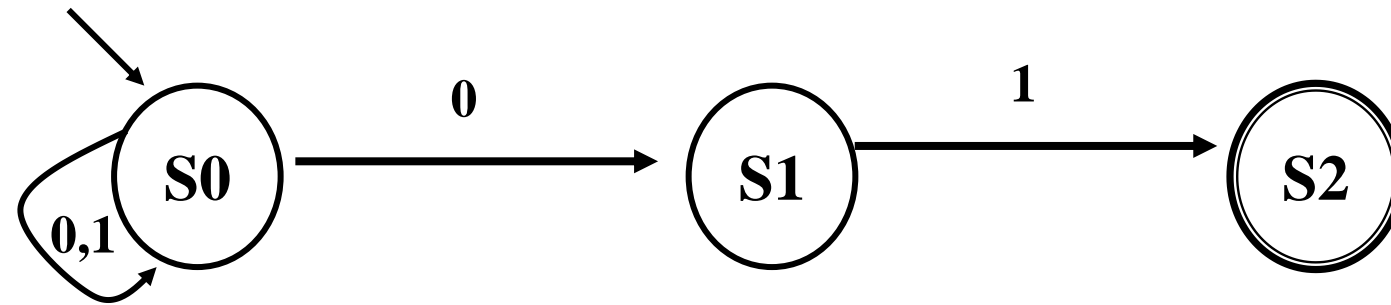
	0	1
S0	{S0,S1}	S0
S1	-	S2
S2	-	-

Exemplo AFND



- ♦ Alfabeto da máquina= $\{0,1\}$.
- ♦ dois arcos como 0 saindo de S0.
- ♦ Também S1 e S2 não possui todos os possíveis arcos do alfabeto.
- ♦ Sempre esquerda para direita.
- ♦ Máquina de reconhecimento?
- ♦ Característica: incompleto para uma máquina de reconhecimento.

Possíveis transições para a entrada 00101

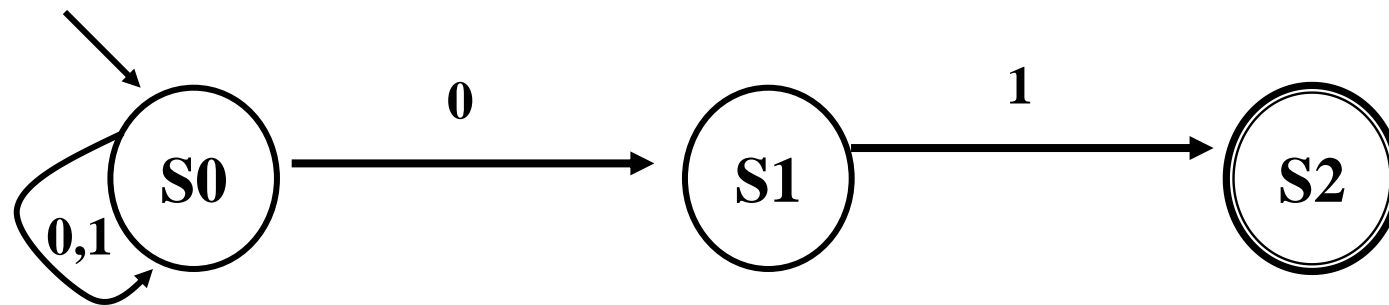


Logo, possibilidades de não finalizar a entrada, não chegar a um estado final e parar

Ideal seria terminar as entradas e atingir o estado final s2

Seta vermelha todas as entradas reconhecidas e chegou em S2.

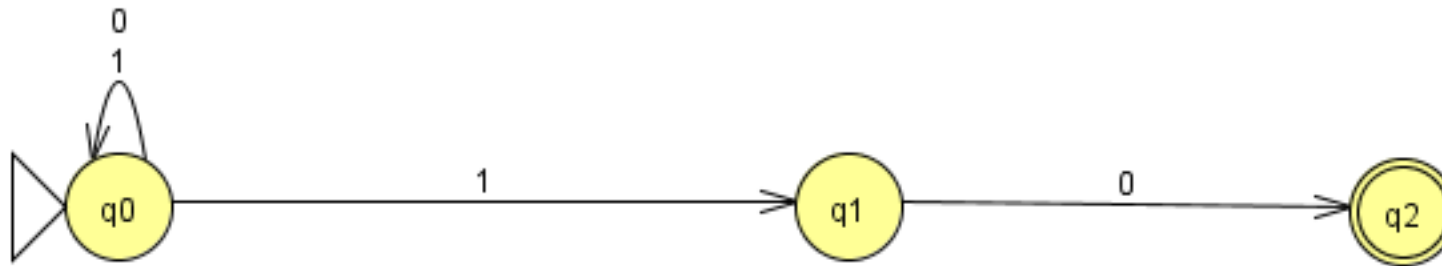
Tabela de possibilidades



		0		0		1		0		1		RESULTADO
1º	S0		S0		S0		S0		S0		S0	passa toda a string mas não chega no estado final.
2º	S0		S0		S1		S2					chega no estado final mas não passa toda a string.
3º	S0		S1									não passa toda a string
4º	S0		S0		S0		S0		S1		S2	passa toda a string e chega ao estado final.

AFND - Autômato Finito não Determinístico

- ◆ Exemplo 1: O exemplo será de um AFND que aceita cadeias de caracteres sobre o alfabeto $\{0, 1\}$ e que contém pelo menos um "1" seguido imediatamente por um "0".



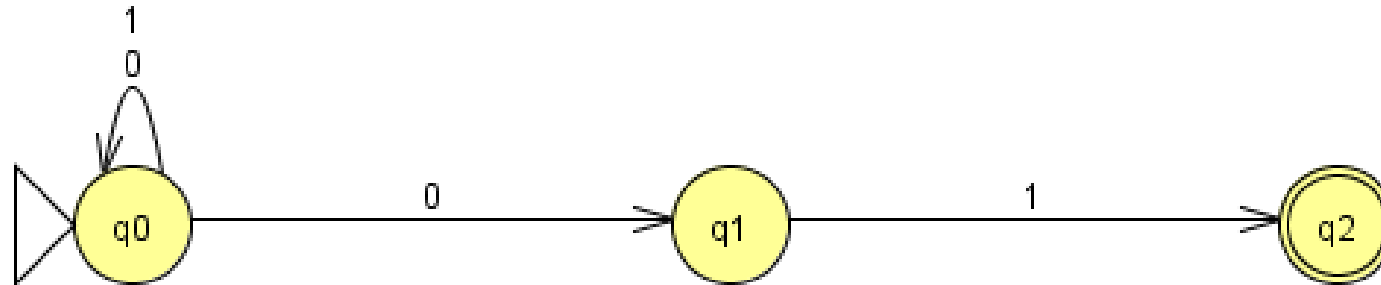
AFND - Autômato Finito não Determinístico

- ◆ Exemplo 1: Exemplos de cadeia aceita:

Input	Result
110	Accept
111	Reject
100	Reject
101	Reject
110110	Accept
011	Reject
001	Reject
010	Accept
000110	Accept

AFND - Autômato Finito não Determinístico

- ◆ Exemplo 2: AFND que Aceita Cadeias Terminadas em "01"



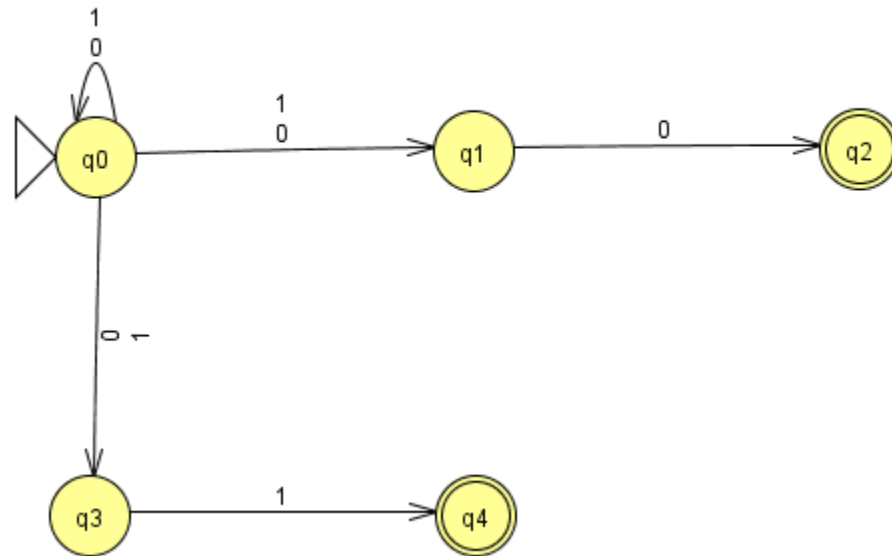
AFND - Autômato Finito não Determinístico

- ◆ Exemplo 2: O AFND aceita cadeias que terminam em "01".

Input	Result
111	Reject
0	Reject
1	Reject
11101	Accept
101	Accept
1101	Accept
11101	Accept
01010101	Accept
11101010101	Accept
11111101	Accept

AFND - Autômato Finito não Determinístico

- ◆ Exemplo 3: AFND que aceita cadeias com "00" ou "11" em qualquer lugar.



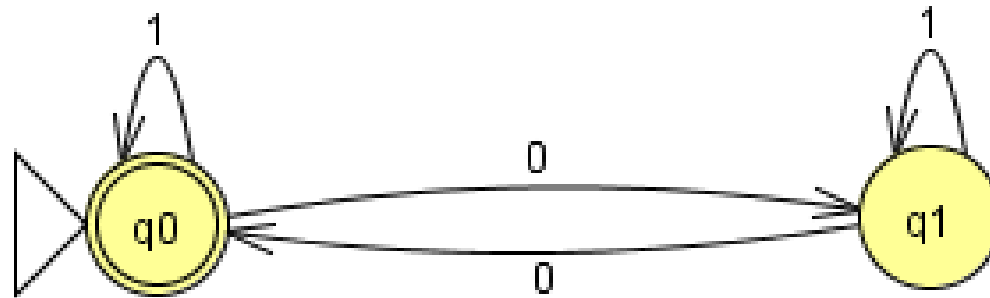
AFND - Autômato Finito não Determinístico

- ◆ Exemplo 3: O AFND aceita qualquer cadeia que contenha "00" ou "11".

Input	Result
00	Accept
11	Accept
0000	Accept
1111	Accept
010101	Accept
101010	Accept
11100	Accept
1100	Accept
0011	Accept
1111010101100	Accept
101010101010	Accept
101010101010101	Accept

AFND - Autômato Finito não Determinístico

- ◆ Exemplo 4: AFND que aceita cadeias com um número par de "0".



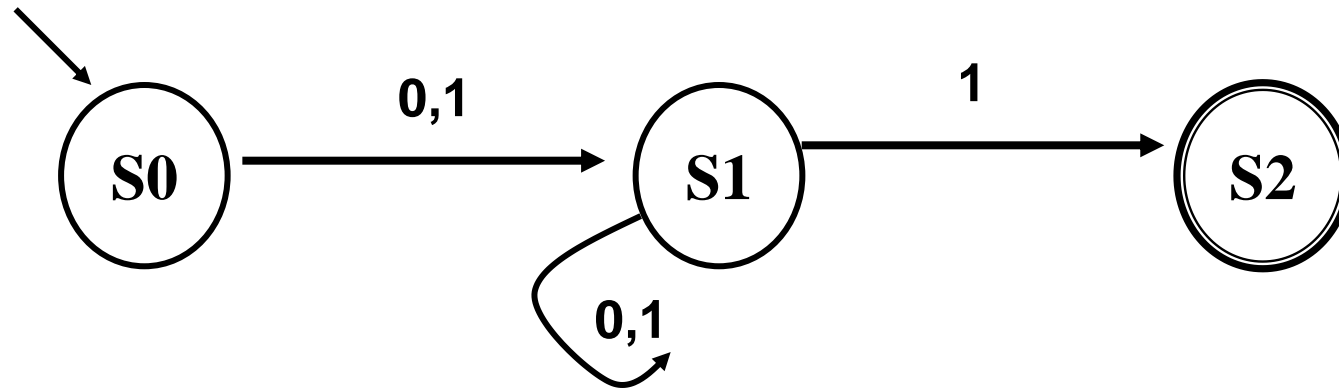
AFND - Autômato Finito não Determinístico

- ◆ Exemplo 4: AFND que aceita cadeias com um número par de "0".

Input	Result
00	Accept
1	Accept
0	Reject
10	Reject
11	Accept
01	Reject
1100	Accept
100	Accept
1100	Accept
11100	Accept
111100	Accept
101010	Reject
010101	Reject
10	Reject
110	Reject

Exercícios

1) Criar a tabela de possibilidades para o AFND.



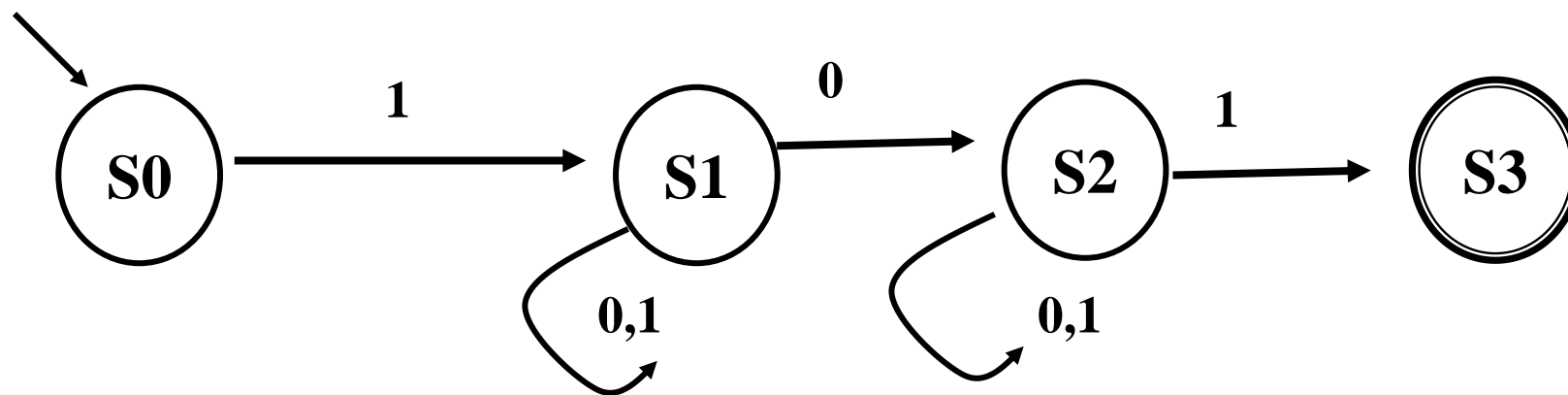
a) 010010

■ S1 S1 S1 S1 S2

b) 111001

Exercícios

2) Criar a tabela de possibilidades para o AFND.

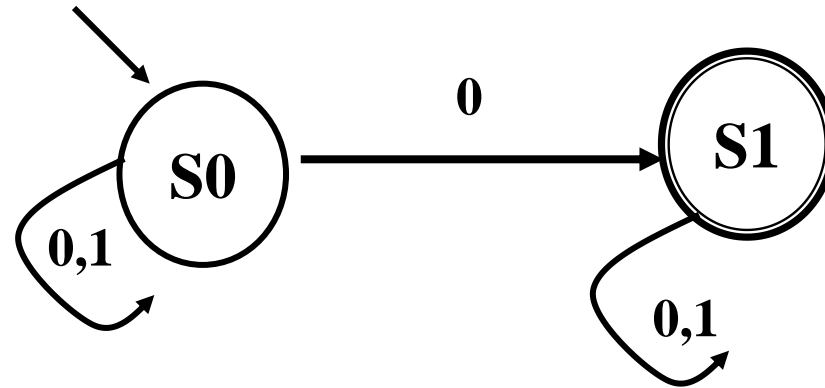


a) 1100

b) 101001

Exercícios

3) Criar a tabela de possibilidades para o AFND.



a) 000110

b) 100110

Transições não determinísticas e suas implicações

- ◆ As transições não determinísticas são um aspecto central dos Autômatos Finitos Não Determinísticos (AFNDs) e têm implicações significativas tanto na teoria quanto na prática da computação.

O que são Transições Não Determinísticas?

- ♦ Uma transição não determinística ocorre quando, a partir de um dado estado e com um dado símbolo de entrada, o autômato pode transitar para um ou mais estados diferentes.
- ♦ Isso difere dos Autômatos Finitos Determinísticos (AFDs), onde cada par de estado e símbolo de entrada corresponde a exatamente uma transição de estado.

Características das Transições Não Determinísticas

- ◆ Múltiplas Opções de Transição:
 - Para um estado e um símbolo de entrada, pode haver várias transições possíveis.

Características das Transições Não Determinísticas

- ◆ Transições ϵ (Epsilon):
 - Alguns AFNDs podem realizar transições de estado sem consumir nenhum símbolo da entrada (transições ϵ).

Características das Transições Não Determinísticas

- ◆ Simultaneidade:
 - Um AFND pode "dividir-se" em múltiplas cópias, cada uma seguindo um caminho de transição diferente ao mesmo tempo.

Implicações das Transições Não Determinísticas

- ◆ Poder de Expressão:
 - Apesar de parecerem mais poderosos, os AFNDs não são mais poderosos do que os AFDs em termos de tipos de linguagens que podem reconhecer (ambos reconhecem linguagens regulares).
 - A diferença está na expressividade e na simplicidade com que certas linguagens podem ser representadas.

Implicações das Transições Não Determinísticas

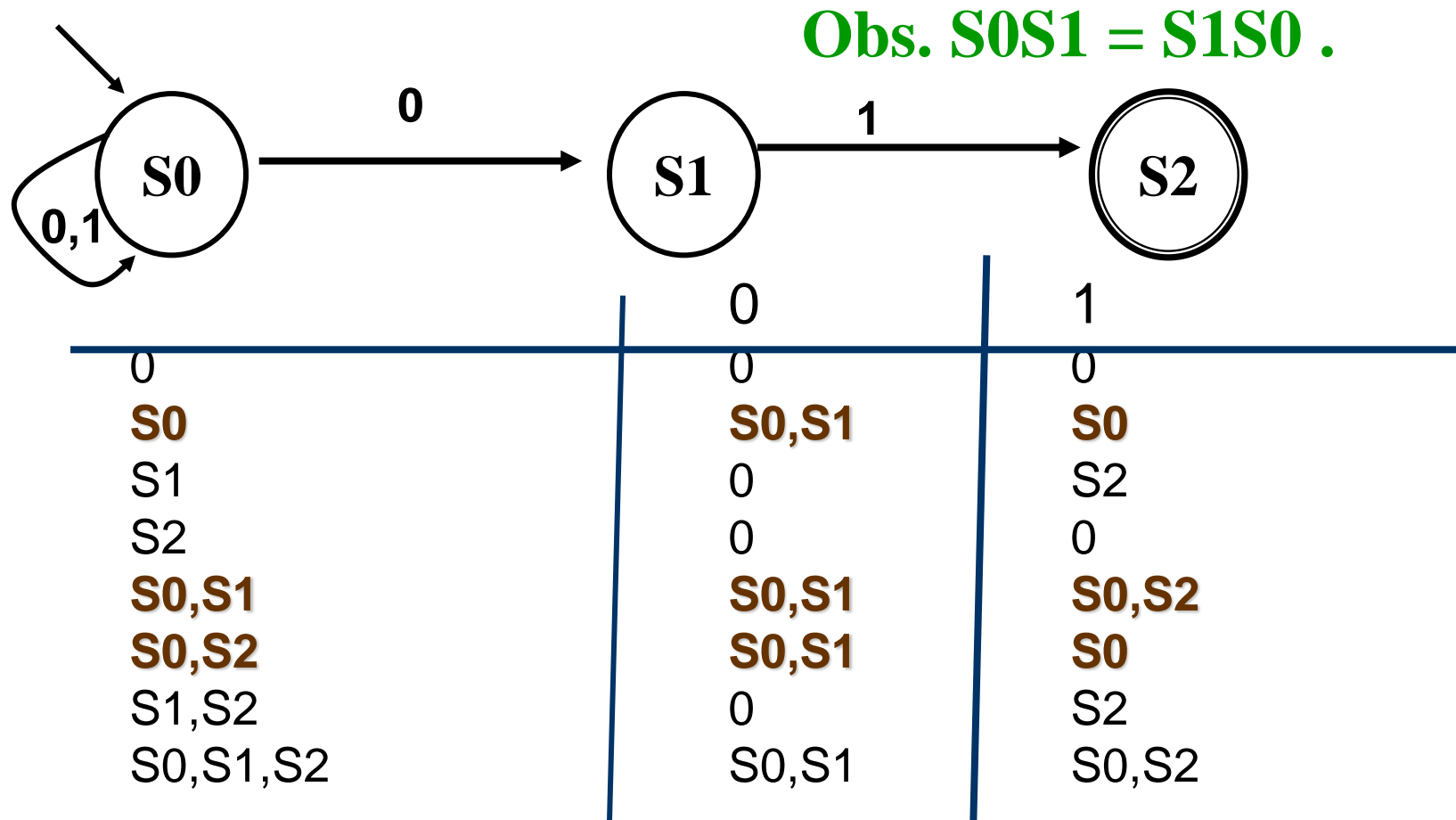
- ◆ Complexidade de Implementação:
 - Implementar um AFND pode ser mais complexo do que um AFD, especialmente porque a simultaneidade das transições não determinísticas pode exigir o rastreamento de múltiplas "cópias" do estado do autômato.
 - Essa complexidade se torna aparente ao converter um AFND em um AFD equivalente, um processo que pode resultar em um aumento exponencial no número de estados.

Conversão AFND para AFD

- ♦ A conversão de um Autômato Finito Não Determinístico (AFND) para um Autômato Finito Determinístico (AFD) é um processo fundamental na teoria da computação.
- ♦ Este processo demonstra que ambos os tipos de autômatos são equivalentes em termos de poder de reconhecimento de linguagens regulares, embora se diferenciem na forma como processam as entradas.
- ♦ Casos que não existe solução, o autômato continua não determinístico e somente é possível testar a máquina através de possibilidades.

Convertendo máquinas AFND em AFD

Todas as possíveis transições



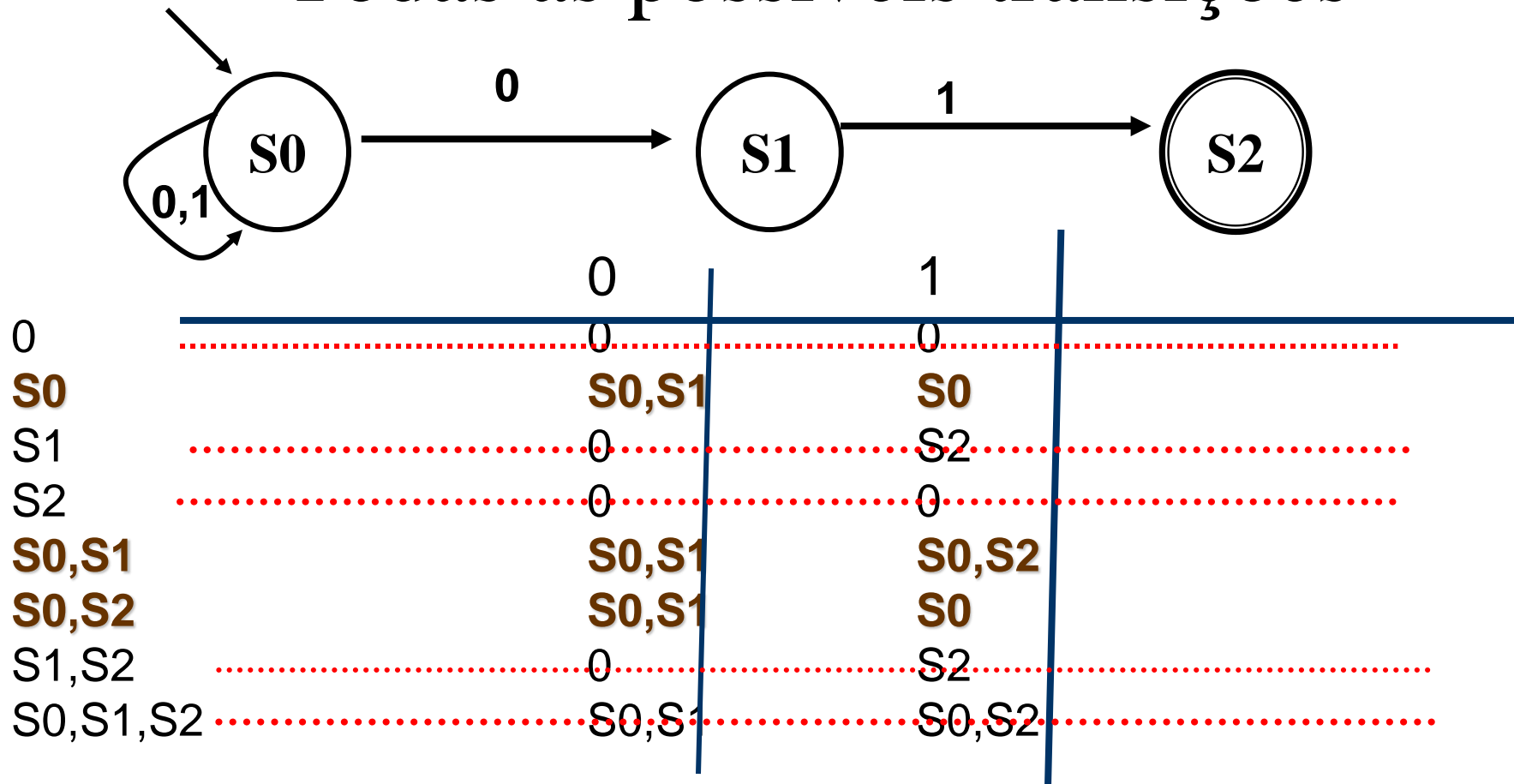
Eliminar estados de acordo com regras...

Regras para eliminar estados

- ◆ 1º Estado com destino 0 (ou - traço).
- ◆ 2º Estado com igual referência, exceto se houver dependência.
- ◆ 3º Estado com possibilidades não alcançáveis (ninguém chama ele).
- ◆ Se depois de eliminar não existe possibilidade de criação do AFD caracterizamos o autômato AFND somente e dada uma entrada, aplicar as possibilidades e resultados (tabela).

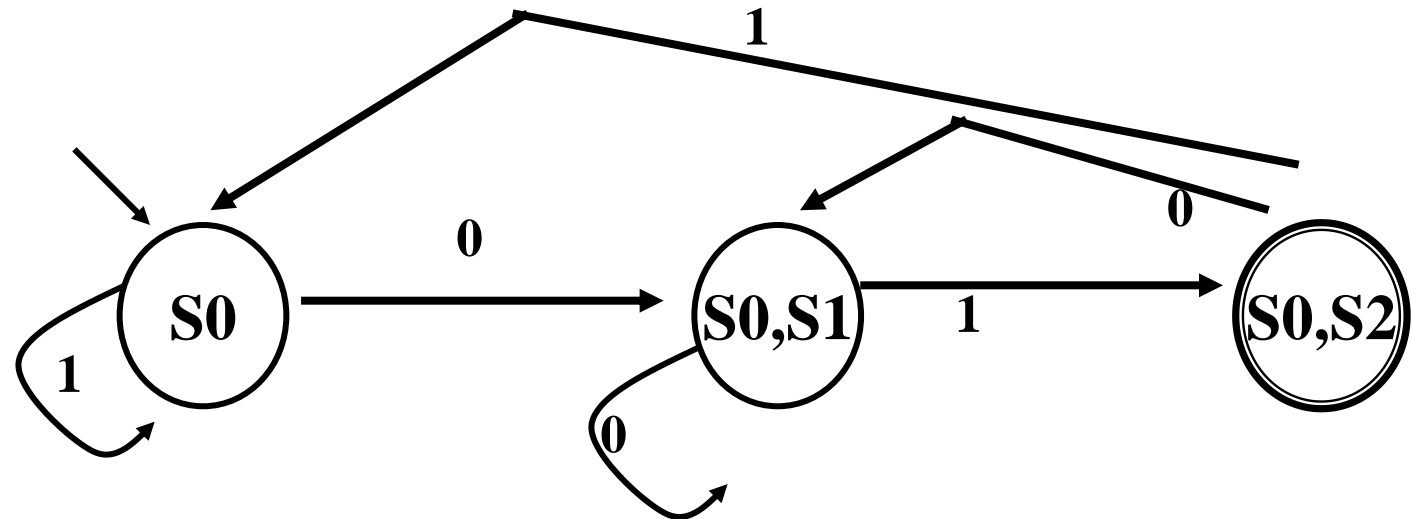
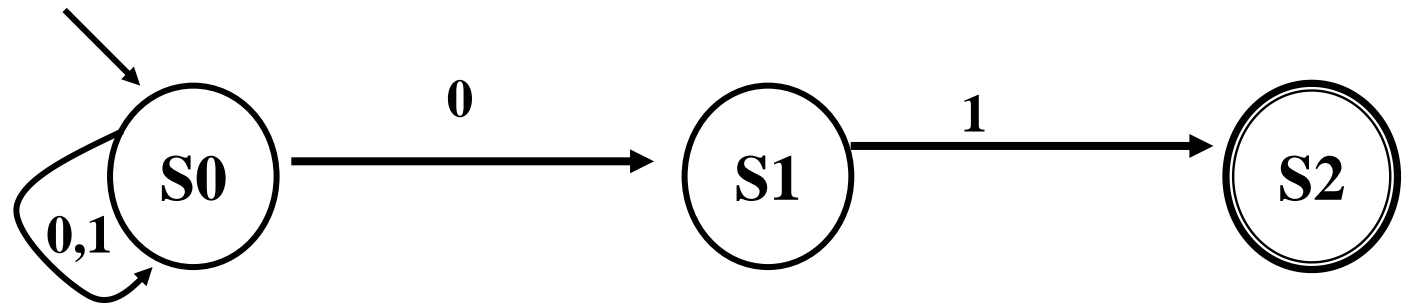
Convertendo máquinas AFND em AFD

Todas as possíveis transições



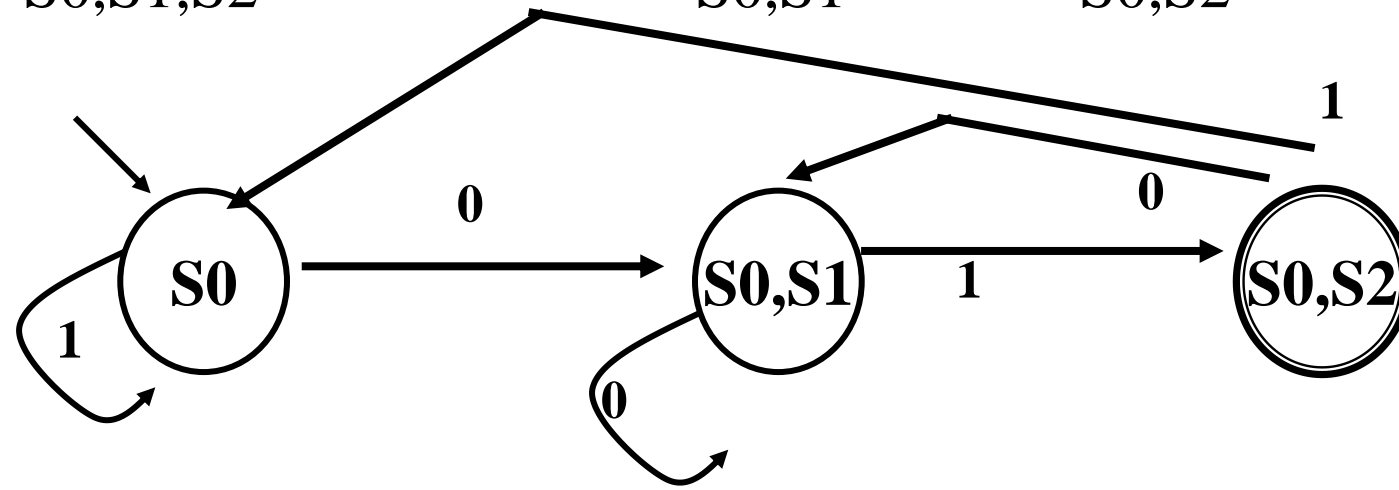
Eliminado pois linha igual acima e
Nenhum estado chama este estado...

Autômato AFD



Autômato AFD

	0	1
0	0	0
S0	S0,S1	S0
S1	0	S2
S2	0	0
S0,S1	S0,S1	S0,S2
S0,S2	S0,S1	S0
S1,S2	0	S2
S0,S1,S2	S0,S1	S0,S2



Ou ainda...

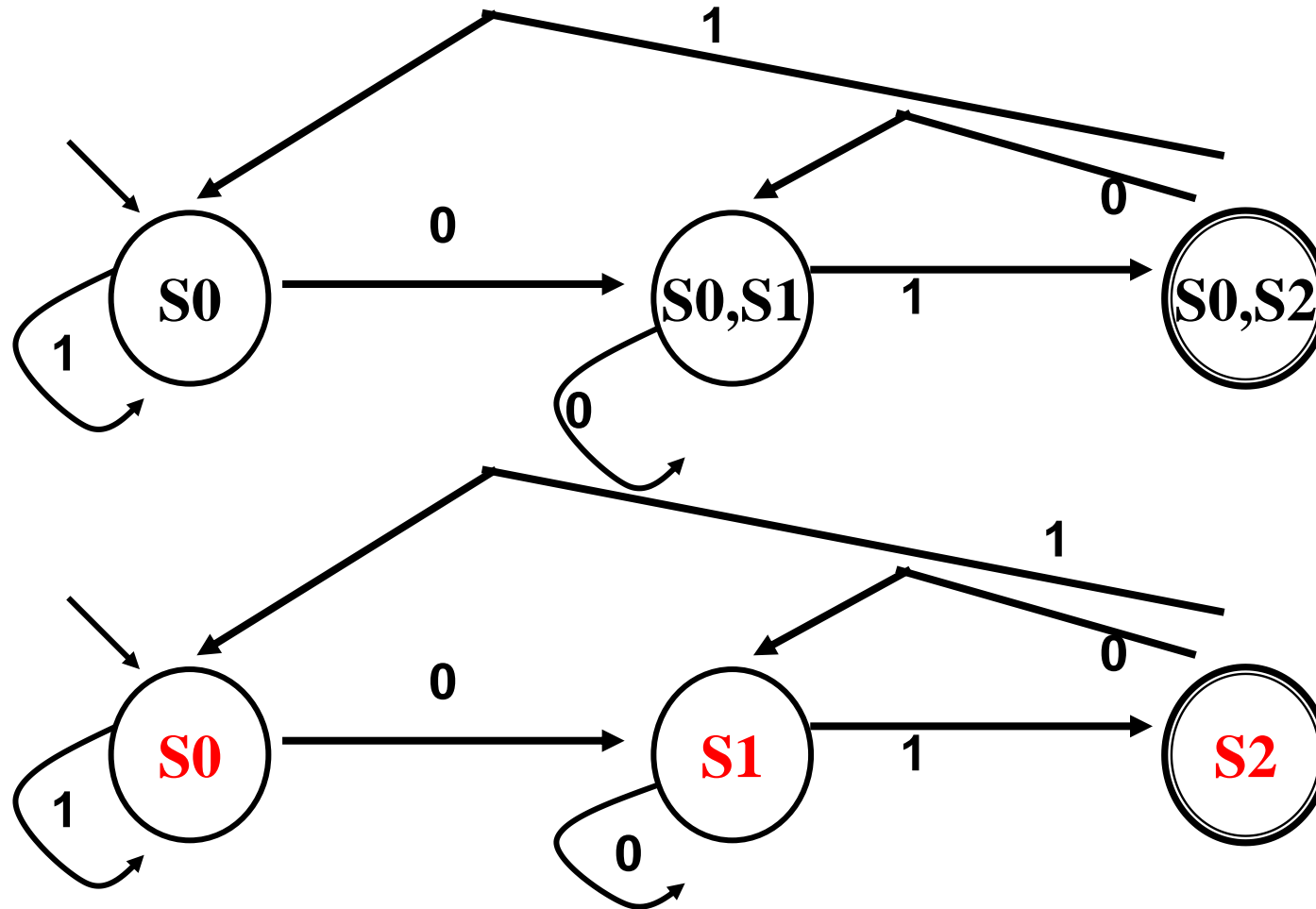
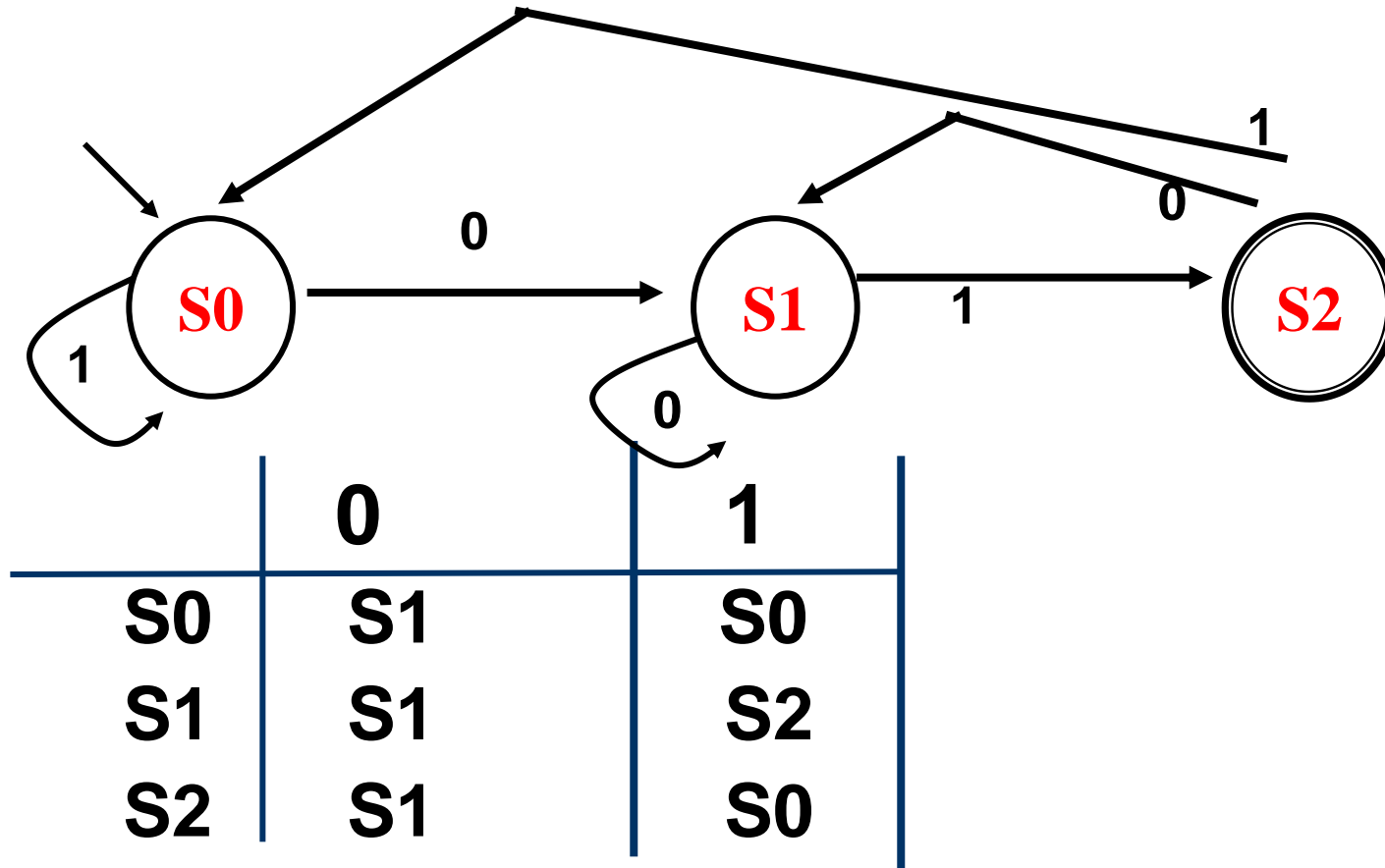
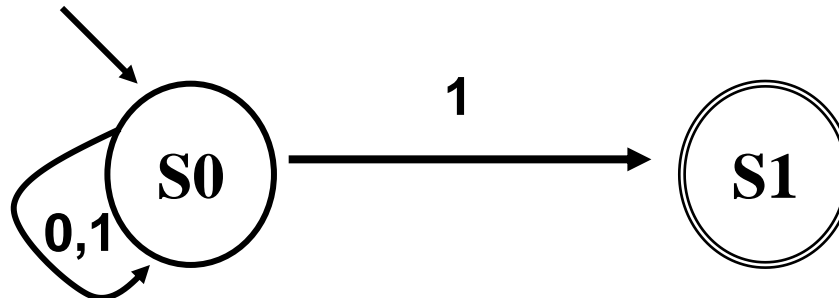


Tabela de transição



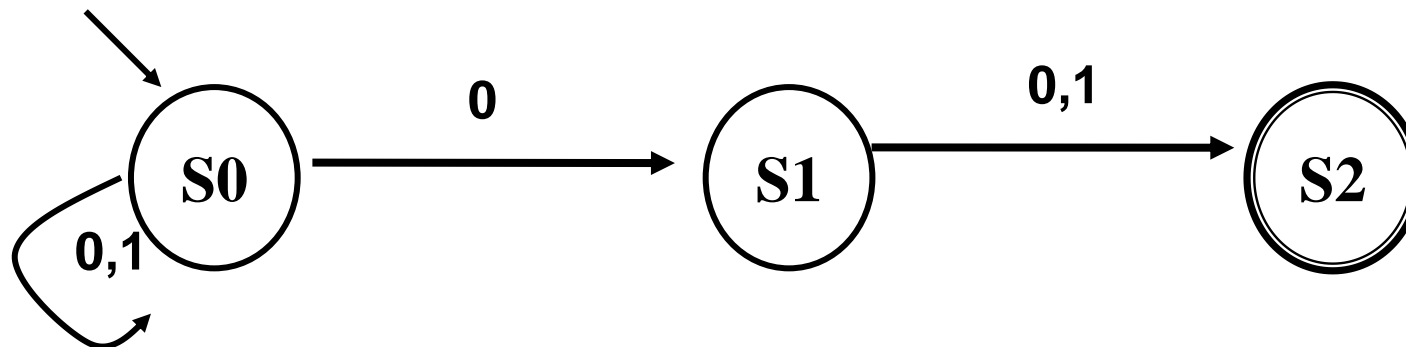
Exercícios 1

- a) Aplicar as regras para eliminar estados.
- b) Transformar o AFND para AFD, se possível.
- c) Apresentar o novo autômato AFD, se possível.
- d) Criar a tabela de estados para o AFD, se possível.



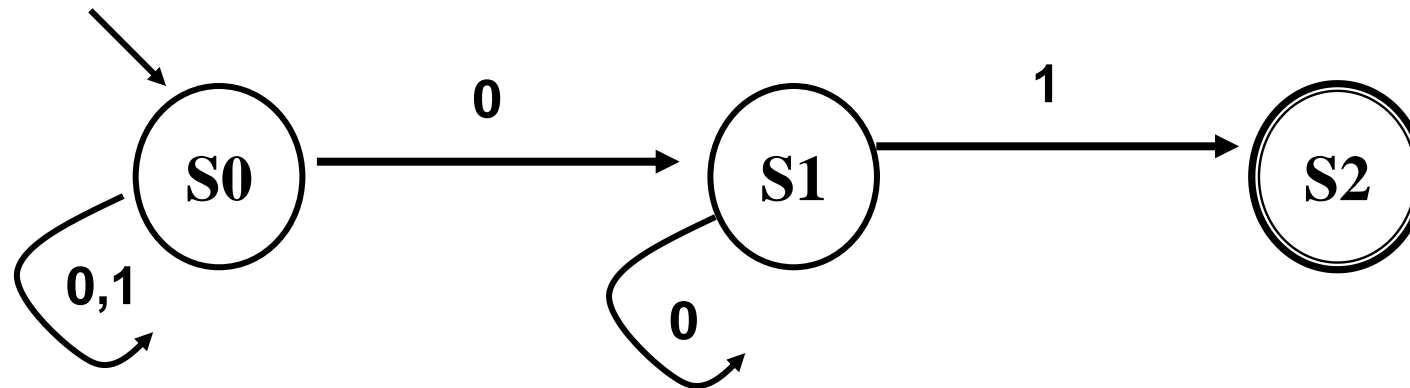
Exercícios 2

- a) Aplicar as regras para eliminar estados.
- b) Transformar o AFND para AFD, se possível.
- c) Apresentar o novo autômato AFD, se possível.
- d) Criar a tabela de estados para o AFD, se possível.



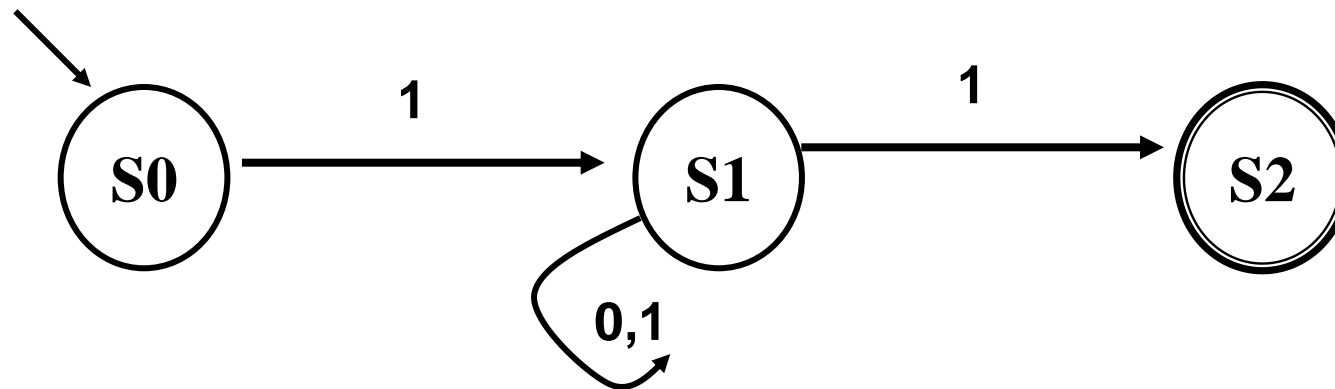
Exercício 3

- a) Aplicar as regras para eliminar estados.
- b) Transformar o AFND para AFD, se possível.
- c) Apresentar o novo autômato AFD, se possível.
- d) Criar a tabela de estados para o AFD, se possível.



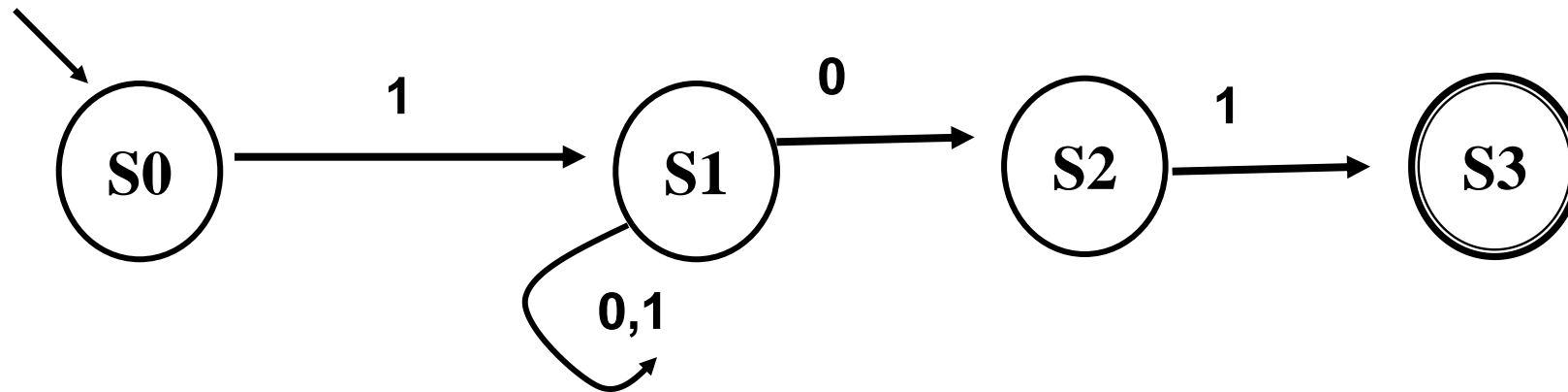
Exercício 4

- a) Aplicar as regras para eliminar estados.
- b) Transformar o AFND para AFD, se possível.
- c) Apresentar o novo autômato AFD, se possível.
- d) Criar a tabela de estados para o AFD, se possível.



Exercício 5

- a) Aplicar as regras para eliminar estados.
- b) Transformar o AFND para AFD, se possível.
- c) Apresentar o novo autômato AFD, se possível.
- d) Criar a tabela de estados para o AFD, se possível.



Referência desta aula

- ◆ HOPCROFT, John E.; MOTWANI, Rajeev; ULLMAN, Jeffrey D. Introdução a teoria de autômatos, linguagens e computação. Rio de Janeiro: Campus, 2002.
- ◆ MENEZES, P. B. Linguagens Formais e Autômatos. Editora SagraLuzzato, 2000.

FIM

Obrigado