Construção de Compiladores

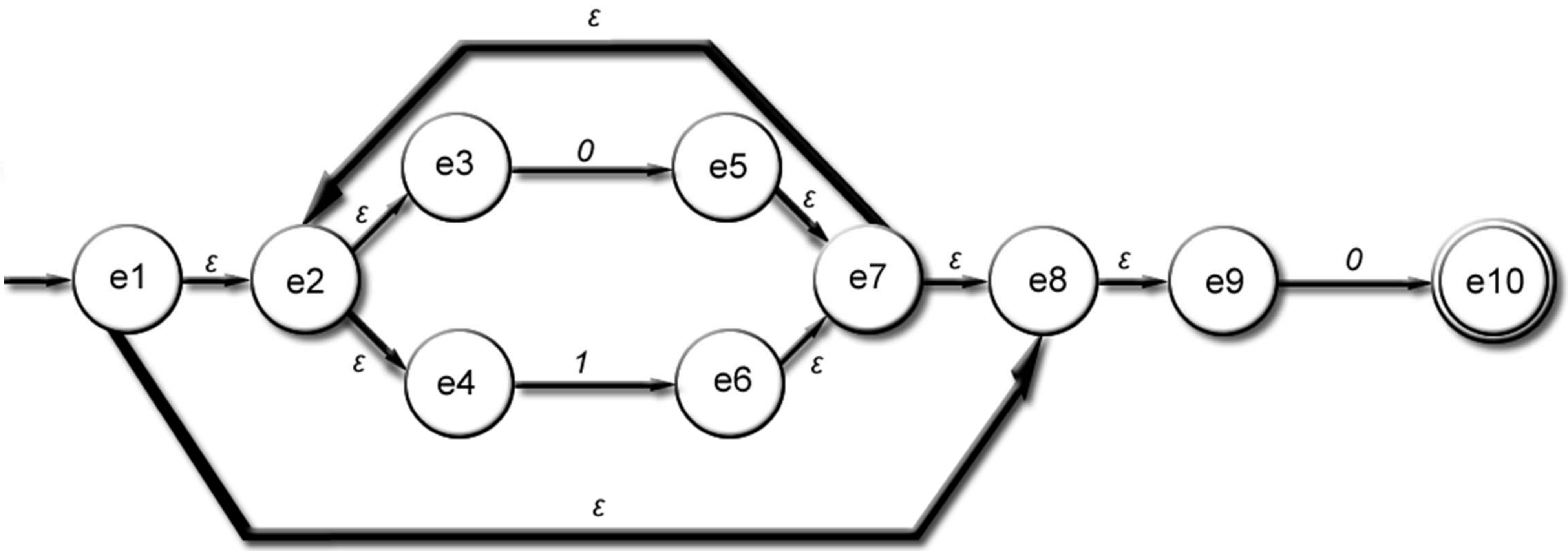
Autômatos

Professor: Luciano Ferreira Silva, Dr.

Aula 5

Algoritmo de Thompson

- O autômato finito não-determinístico para reconhecer a expressão (0|1)*0 é:



Conversão para autômatos determinísticos

- **Autômatos não-determinísticos possuem diversas situações de ambigüidade como:**
  - ✓ Ao partir de e1, por que o autômato segue para e2 e não para e8?
  - ✓ Por que a transição seguinte foi de e2 para e3 e não para e4?
- **Solução: Conversão para autômatos determinísticos**

Conversão para autômatos determinísticos

- **Procedimento sistemático: Método da construção de subconjuntos;**
  - ✓ Criação e associação de novos estados do autômato determinístico com conjuntos de estados do autômato não-determinístico;
  - ✓ Definição de subconjuntos de estados do autômato não-determinístico por meio do operador ε*(épsilon-clausura);
    - Sua aplicação resulta no conjunto que inclui, além dos próprios estados, cada um dos demais estados do autômato que podem ser alcançados a partir desses com transições pela string vazia.

Conversão para autômatos determinísticos

- **Inicia-se pelo estado inicial do autômato não-determinístico;**

  ✓ ε*{e1}={e1, e2, e3, e4, e8, e9}

  ✓ Esse subconjunto é associando ao estado inicial do autômato finito determinístico, que pode receber o nome de S_0;

Conversão para autômatos determinísticos

Analisando S_0;

S_0	e1	e2	e3	e4	e8	e9
0	—	—	e5	—	—	e10
1	—	—	—	e6	—	—

✓**0** levará S_0 a um estado que corresponderá a

$\varepsilon^*\{e5, e10\}=\{e2, e3, e4, e5, e7, e8, e9, e10\}$ (S_1)

✓**1** levará S_0 a um estado que corresponderá a

$\varepsilon^*\{e6\} =\{e2, e3, e4, e6, e7, e8, e9\}$ (S_2)

Conversão para autômatos determinísticos

- **Observações:**

✓ $S_0 \neq S_1 \neq S_2$

✓ Caso um destes conjuntos fosse igual a outro não seria viabilizada sua construção;

✓ S_1 contém e10 (estado final do autômato não-determinístico), portanto S_1 é um estado final.

Conversão para autômatos determinísticos

Analisando S_1;

S_1	e_2	e_3	e_4	e_6	e_7	e_8	e_9	e_{10}
0	—	e_5	—	—	—	—	e_{10}	—
1	—	—	e_6	—	—	—	—	—

✓**0** levará S_1 a um estado que corresponderá a

ε*{e5, e10 }(ou seja, o próprio S_1);

✓**1** levará S_1 a um estado que corresponderá a

ε*{e6} (que é S_2)

Conversão para autômatos determinísticos

▪ Analisando S_2;

S_2	e_2	e_3	e_4	e_6	e_7	e_8	e_9
0	—	e_5	—	—	—	—	e_{10}
1	—	—	e_6	—	—	—	—

✓**0** levará S_1 a um estado que corresponderá a ε*{e5, e10 } (que é S_1);

✓**1** levará S_1 a um estado que corresponderá a ε*{e6} (ou seja, o próprio S_2);

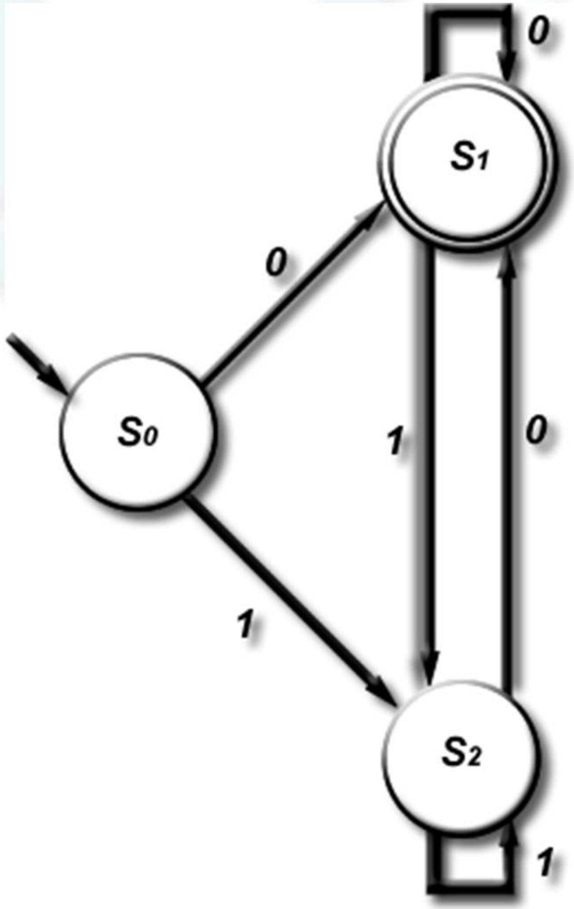
Conversão para autômatos determinísticos

- Portanto o autômato determinístico para reconhecer (0|1)*0 é dado pela matriz:

	S_0	S_1	S_2
0	S_1	S_1	S_1
1	S_2	S_2	S_2

Estado inicial: S_0

Estados finais: S_1



Minimização de estados do autômato

- Consiste em combinar estados redundantes do autômato em um único estado sem alterar a linguagem que é reconhecida;
- Procedimento: construção iterativa de partições do conjunto K de estados do autômato.

Minimização de estados do autômato

- **Primeira partição (P_1)**
  - ✓Separa os estado finais dos não-finais;
  - ✓$P_1=\{C_1, C_2\}$, onde $C_1 = F$ e $C_2 = K – F$;
- **Para o autômato que reconhece (0|1)*0 tem-se:**

$$P_1 = \{C_1, C_2\}$$

$$C_1 = \{S_1\}$$

$$C_2 = \{S_0, S_2\}$$

Minimização de estados do autômato

- C_1 é unitário, portanto S_1 é não redundante;
- C_2 não é unitário, deve ser analisado;

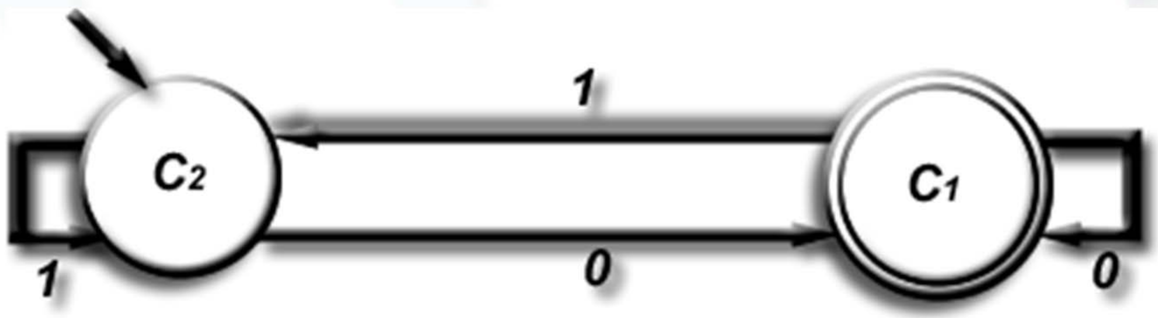
	S_0	S_2
0	C_1	C_1
1	C_2	C_2

- S_0 e S_2 apresentam comportamentos iguais;
- O conjunto C_2 não admite mais partições

Minimização de estados do autômato

- Sendo assim S_0 e S_2 podem ser unidos em um só estado;

- Deste modo tem-se o seguinte autômato:



Minimização de estados do autômato

- **Considere o autômato finito que reconhecer a expressão regular *a*abb****

  ✓ construído com a aplicação do algoritmo de Thompson e com o método da construção de subconjuntos.

	S_0	S_1	S_2	S_3
a	S_1	S_1	—	—
b	—	S_2	S_3	S_3

Estado inicial: S_0

Estados finais: S_2, S_3

Minimização de estados do autômato

- **Tem-se então:**
  - ✓ $P_1 = \{C_1, C_2\}$, com $C_1 = \{S_0, S_1\}$ e $C_2 = \{S_2, S_3\}$
- **Analisando C_1**

	S_0	S_1
a	C_1	C_1
b	—	C_2

- **As colunas são distintas logo na próxima iteração S_0 e S_1 não estarão na mesma partição.**

Minimização de estados do autômato

- **Analisando C_2**

	S_2	S_3
a	—	—
b	C_2	C_2

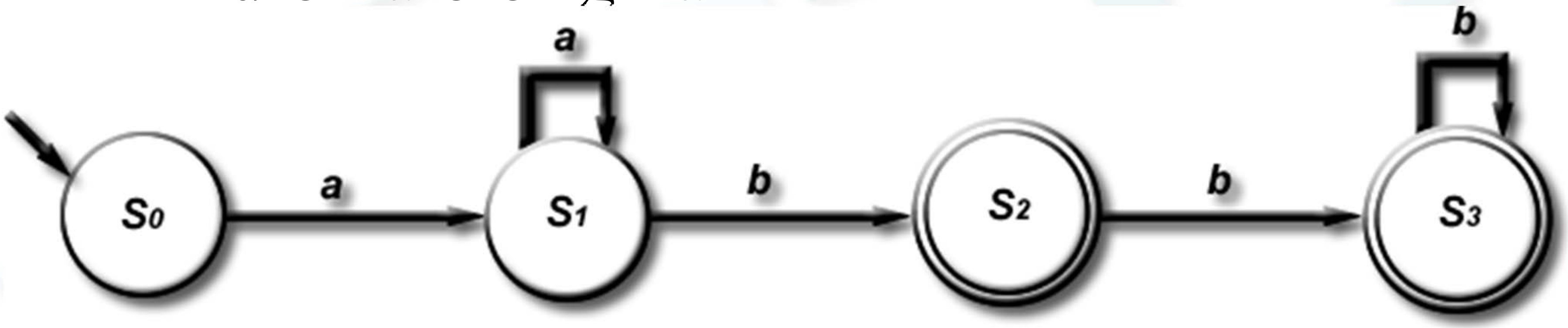
- **S_2 e S_3 apresentam colunas iguais, portanto são redundante e serão unidos;**

- **A próxima partição $P_2 = \{\{S_0\}, \{S_1\}, \{S_2, S_3\}\}$ é a final.**

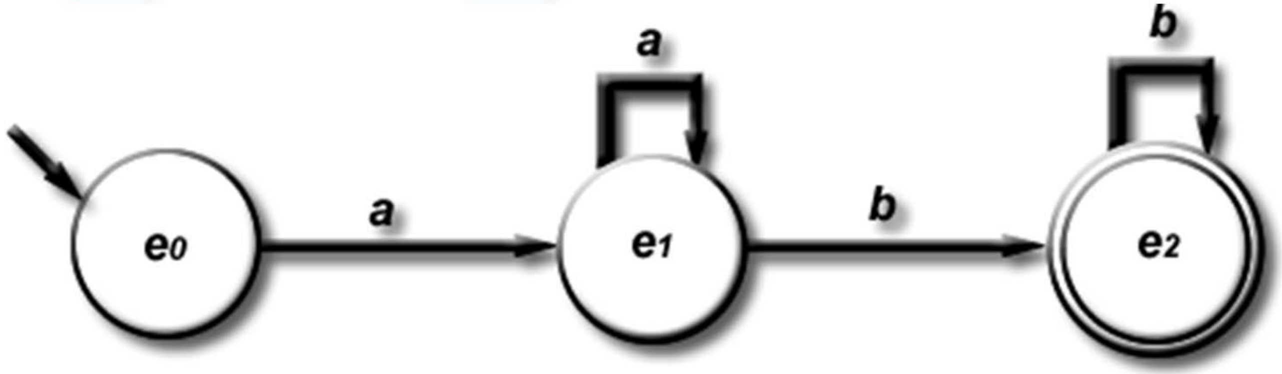
Minimização de estados do autômato

- **O resultado da minimização seria:**

✓Autômato original



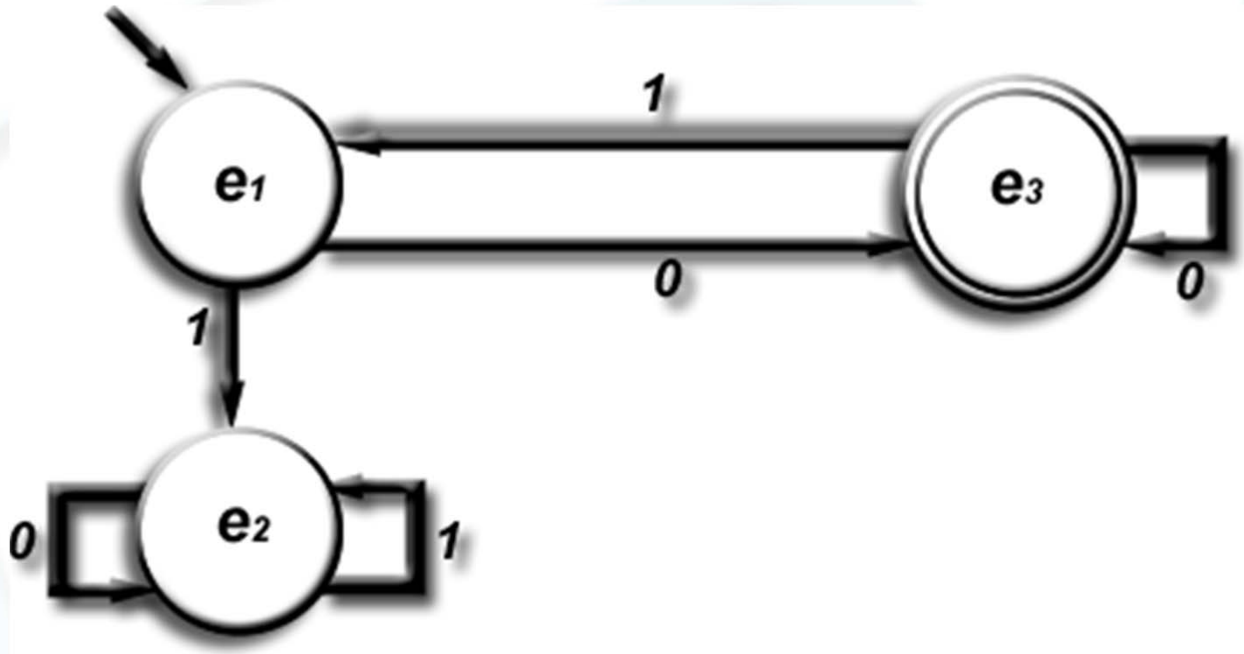
✓Autômato minimizado



UFRR – Departamento de Ciência da Computação
Construção de Compiladores – Prof. Dr. Luciano F. Silva

Minimização de estados do autômato

- Considere o autômato abaixo, ele possui um “estado morto” (e2).



- Portanto este autômato poderia apresentar uma forma mais otimizada, e2 poderia ser eliminado.