Teoria da Computação

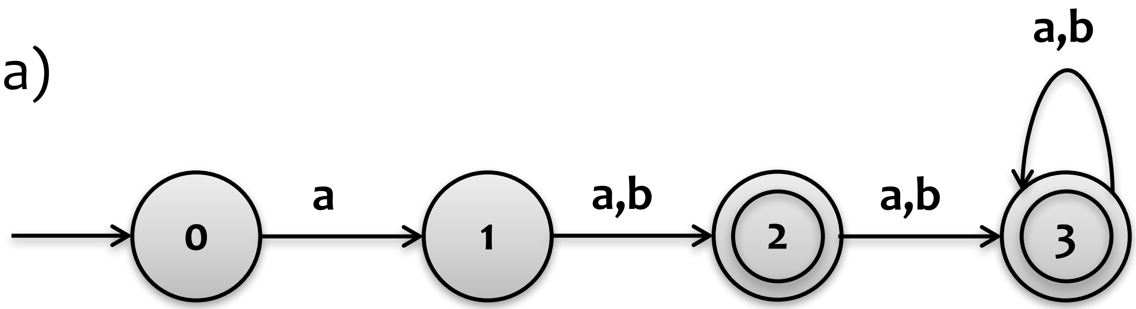
Minimização de AFD's

Material extraído do livro e slides do Prof. Newton Vieira
(http://dcc.ufmg.br/~nvieira)

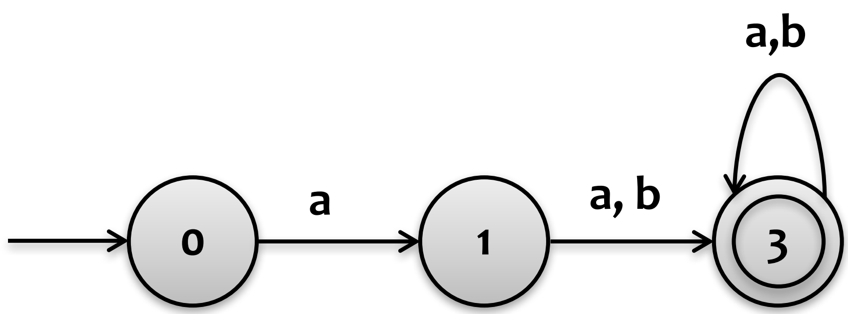
Introdução

- Considere a palavra $w = ababababa$. Em termos de processamento, qual dos autômatos abaixo é mais eficiente?

a)



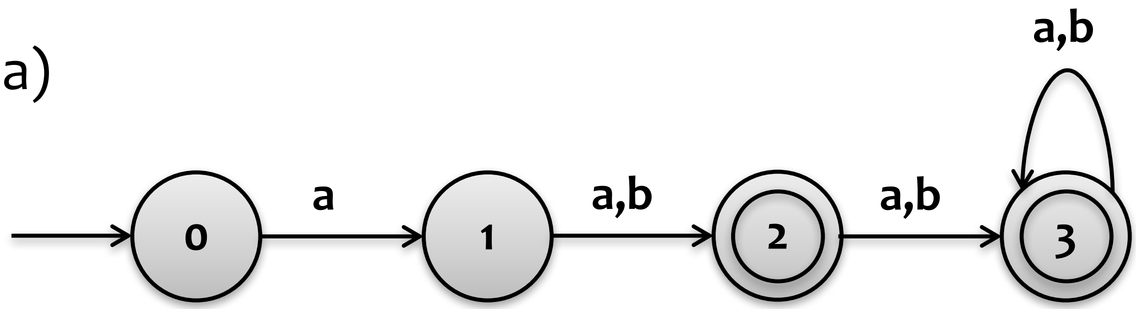
b)



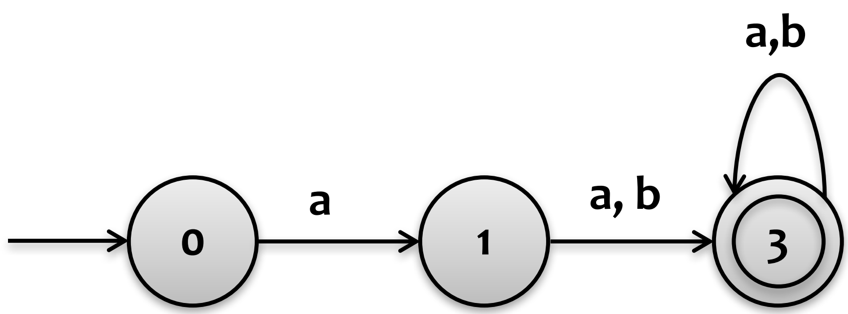
Introdução

- Considere a palavra $w = ababababa$. Em termos de processamento, qual dos autômatos abaixo é mais eficiente?

a)



b)



Por quê minimizar?

Os dois possuem a mesma eficiência, já que o processamento depende do tamanho da palavra e não do tamanho do autômato

AFD Mínimo

- Um AFD M é dito ser **mínimo** para a linguagem $L(M)$ se nenhum AFD para $L(M)$ contém menor número de estados que M

- Para obter um AFD mínimo deve-se
  a) Eliminar **estados não alcançáveis** a partir do estado inicial
  b) Substituir cada grupo de **estados equivalentes** por um único estado

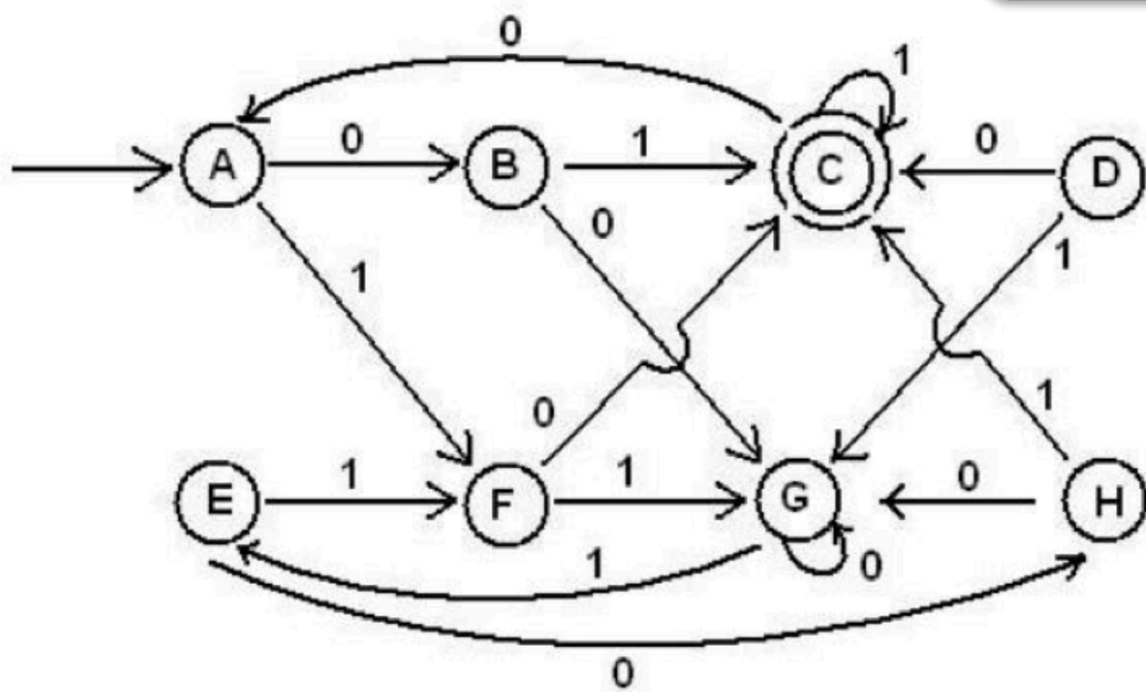
O que são estados equivalentes?

Estados Não Alcançáveis

- Para eliminar estados não alcançáveis pode-se utilizar qualquer algoritmo de busca em grafos, como por exemplo, **busca em profundidade**

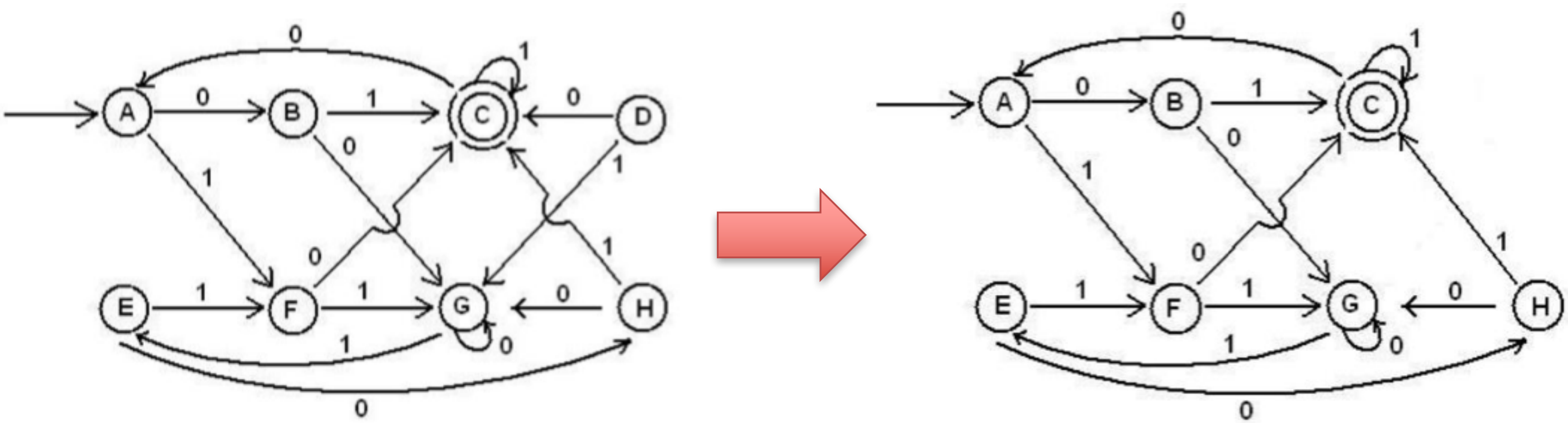
- Exemplo

Quais estados são inalcançáveis?



Estados Não Alcançáveis

- Autômato sem estados não alcançáveis



Estados Equivalentes

- Seja um AFD $M = (E, \Sigma, \delta, i, F)$ Então $e, e' \in E$ são ditos **equivalentes,**$e \approx e'$, se e somente se,

$$\forall y \in \Sigma^*, \hat{\delta}(e, y) \in F \leftrightarrow \hat{\delta}(e', y) \in F$$

A relação ≈ é de equivalência, já que é reflexiva, simétrica e transitiva

- Determinar grupos de estados equivalentes e substituir cada cada grupo por um único estado

Porque reduzir estados equivalentes a um só?

Estados Equivalentes

- Seja um AFD $M = (E, \Sigma, \delta, i, F)$
  - Se $\boldsymbol{e} \approx \boldsymbol{e}'$: um sufixo y é reconhecido passando-se por e se, e somente se, ele é reconhecido passando por e'; logo e e e' podem se tornar um só
  - Se $\boldsymbol{e} \not\approx \boldsymbol{e}'$ $(\exists y \in \Sigma^*, \hat{\delta}(e, y) \in F$ e $\hat{\delta}(e', y) \notin F$ ou vice-versa): se um sufixo y levar a um estado de F, a palavra é aceita, caso contrário, não é. Logo, e e e' não podem se tornar um só

$[e] = \{e_1, e_2, \ldots, e_n\}$ é a classe de equivalência de e na partição induzida por $\approx$

AFD Reduzido

- Seja um AFD $M = (E, \Sigma, \delta, i, F)$, um AFD **reduzido** correspondente a M é o AFD $M' = (E', \Sigma, \delta', i', F')$
  - $E' = \{[e] \mid e \in E\}$
  - $\delta'([e], a) = [\delta(e, a)] \quad \forall e \in E, \forall a \in \Sigma$
  - $i' = [i]$
  - $F' = \{[e] \mid e \in F\}$

O AFD M' é equivalente ao AFD M? Como mostrar isso?

AFD Reduzido

- Para mostrar que um AFD M' reduzido é equivalente ao AFD M, basta mostrar que

a) O processamento é equivalente

$$\hat{\delta}'([e],w) = [\delta(e,w)] \quad \forall w \in \Sigma^* \text{ por indução sobre } |w|$$

a) Reconhecem a mesma linguagem ($L(M') = L(M)$)

$$\hat{\delta}'(i',w) \in F' \leftrightarrow \hat{\delta}(i,w) \in F \quad \forall w \in \Sigma^*$$

O problema do algoritmo de minimização é encontrar as classes de equivalências induzidas pela relação ≈

Classes de Equivalência Induzidas por ≈

- A relação ≈ pode ser definida como uma série de refinamentos ($\approx_0, \approx_1, \approx_2 \ldots$), em que $\approx_{n+1}$ refina $\approx_n$

- A definição de $\approx_i$ $(i \geq 0)$ para um AFD $M = (E, \Sigma, \delta, i, F)$ é dada da seguinte forma

  a) $e \approx_0 e'$ se, e somente se, $e, e' \in F$ ou $e, e' \in E - F$

  b) $e \approx_{n+1} e'$ $(n \geq 0)$ se, e somente se, $e \approx e'$ e

  $\delta(e, a) \approx_n \delta(e', a) \quad \forall a \in \Sigma$

Classes de Equivalência Induzidas por ≈

- A definição anterior pode ser reformulada usando a notação $[e]_n$, onde ela é usada para denotar a classe de equivalência que pertence o estado *e* na partição induzida por $\approx_n$

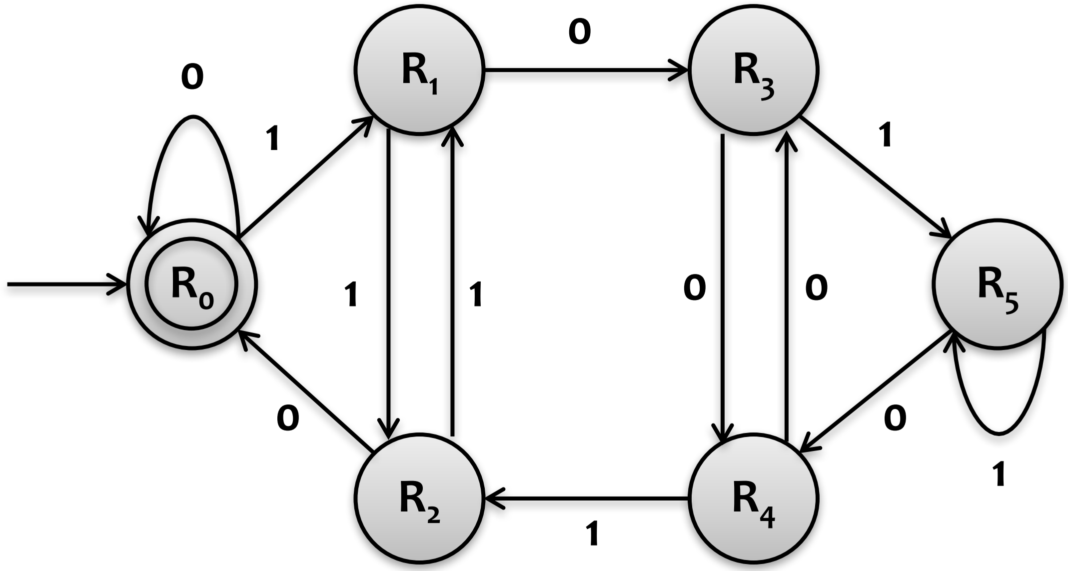
- A definição de $[e]_i$ $(i \geq 0)$ para um AFD $M = (E, \Sigma, \delta, i, F)$ é dada da seguinte forma

a) $[e]_0 = \begin{cases} F & \text{se } e \in F \\ E - F & \text{se } e \in E - F \end{cases}$

b) $[e]_{n+1} = \{e' \in [e]_n \mid [\delta(e', a)]_n = [\delta(e, a)]_n, \forall a \in \Sigma\}$ $(n \geq 0)$

Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)



O objetivo é particionar o conjunto de estados em conjuntos de equivalência

Exemplo

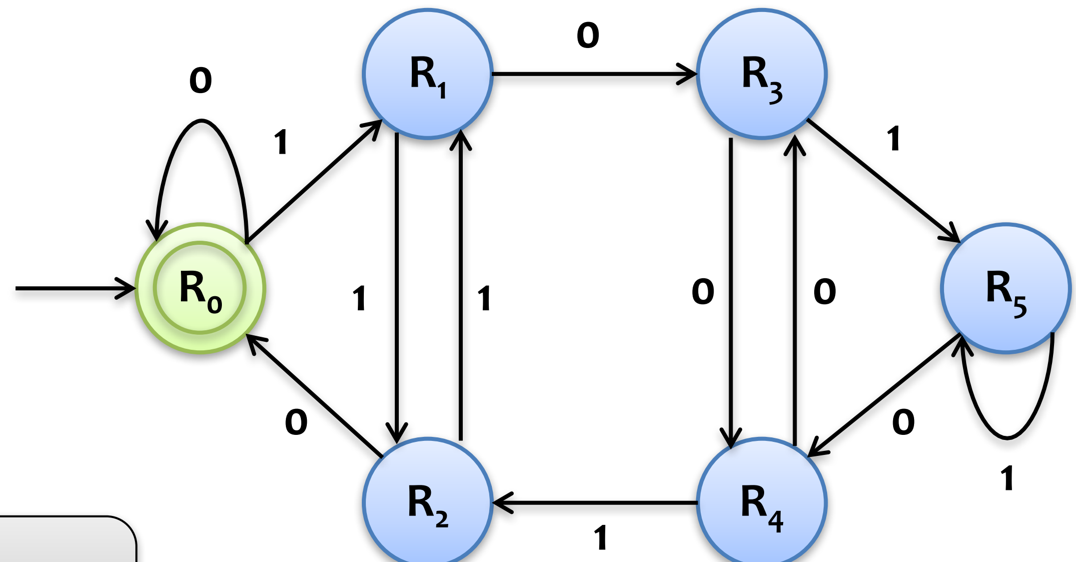
- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_1, R_2, R_3, R_4, R_5\}$

Partição dos estados finais e não-finais



As transições de cada estado levam a qual grupo?

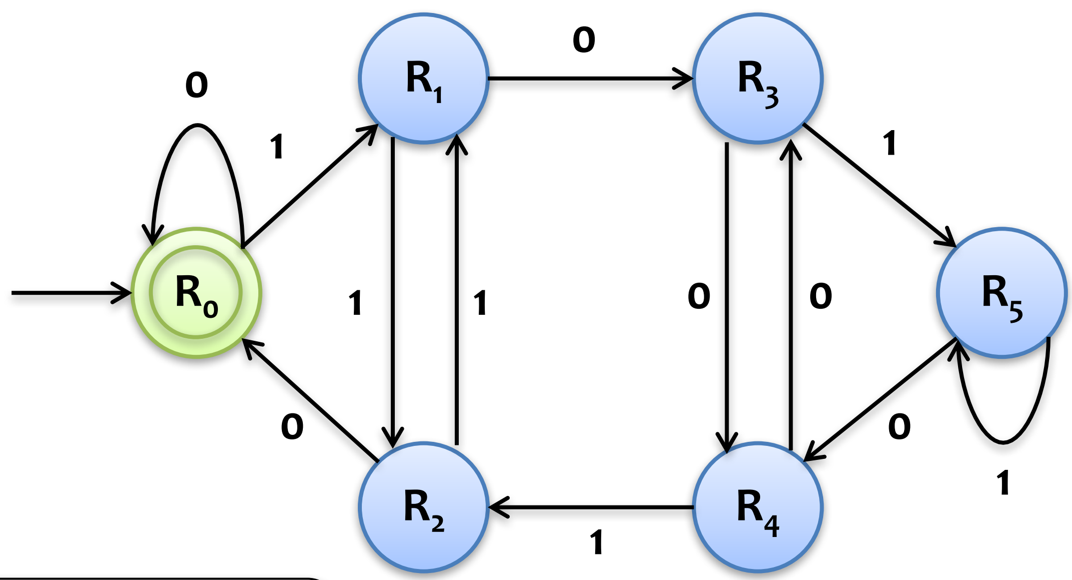
Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_1, R_2, R_3, R_4, R_5\}$



	0	1
R_0	G_1	G_2
R_1	G_2	G_2
R_2	G_1	G_2
R_3	G_2	G_2
R_4	G_2	G_2
R_5	G_2	G_2

Quais os grupos equivalentes?

Exemplo

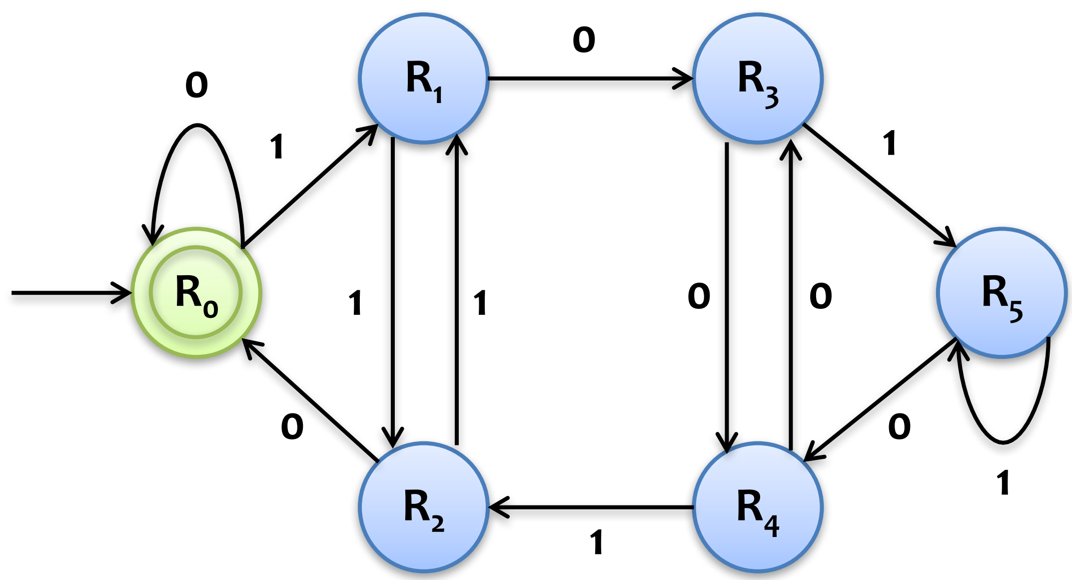
- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_1, R_2, R_3, R_4, R_5\}$

		0	1
G_1	R_0	G_1	G_2
G_2	R_1	G_2	G_2
G_3	R_2	G_1	G_2
G_2	R_3	G_2	G_2
	R_4	G_2	G_2
	R_5	G_2	G_2



Exemplo

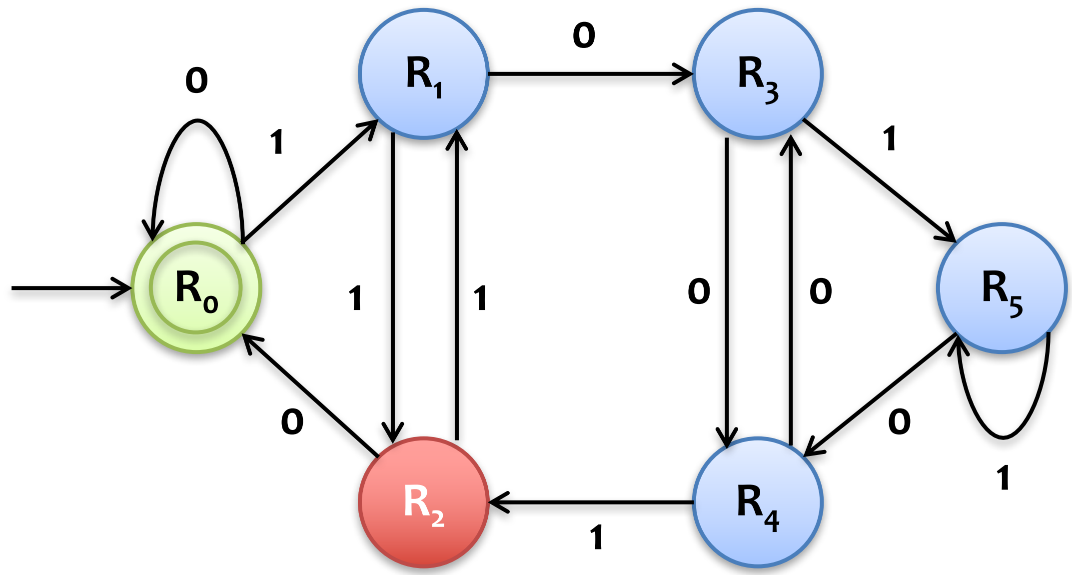
- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_1, R_3, R_4, R_5\}$

$G_3 = \{R_2\}$



Qual o próximo particionamento?

Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

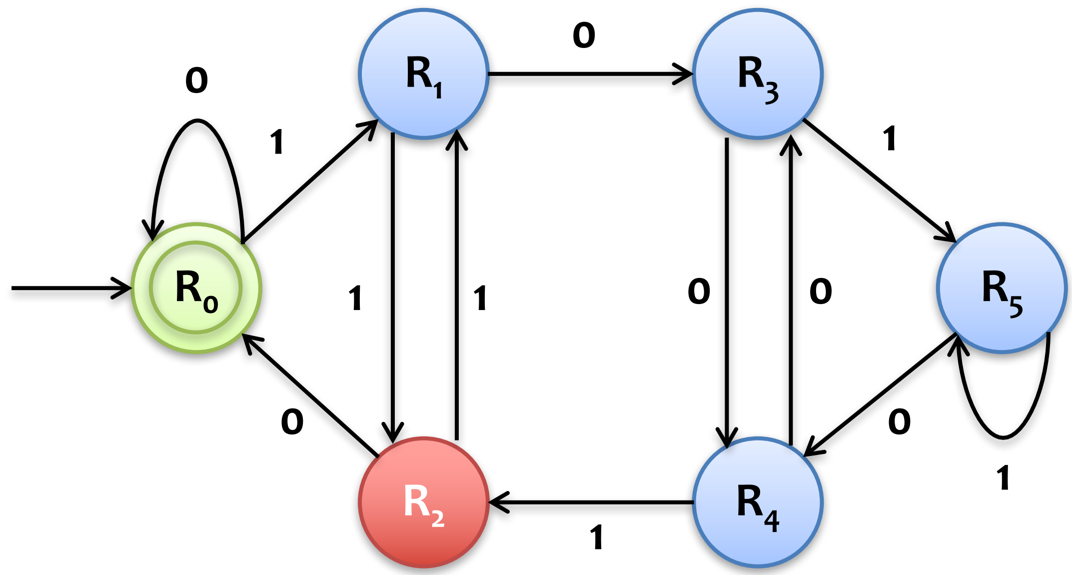
Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_1, R_3, R_4, R_5\}$

$G_3 = \{R_2\}$

		0	1
G_1	R_0	G_1	G_2
G_4	R_1	G_2	G_3
G_3	R_2	G_1	G_2
G_2	R_3	G_2	G_2
G_4	R_4	G_2	G_3
G_2	R_5	G_2	G_2



Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

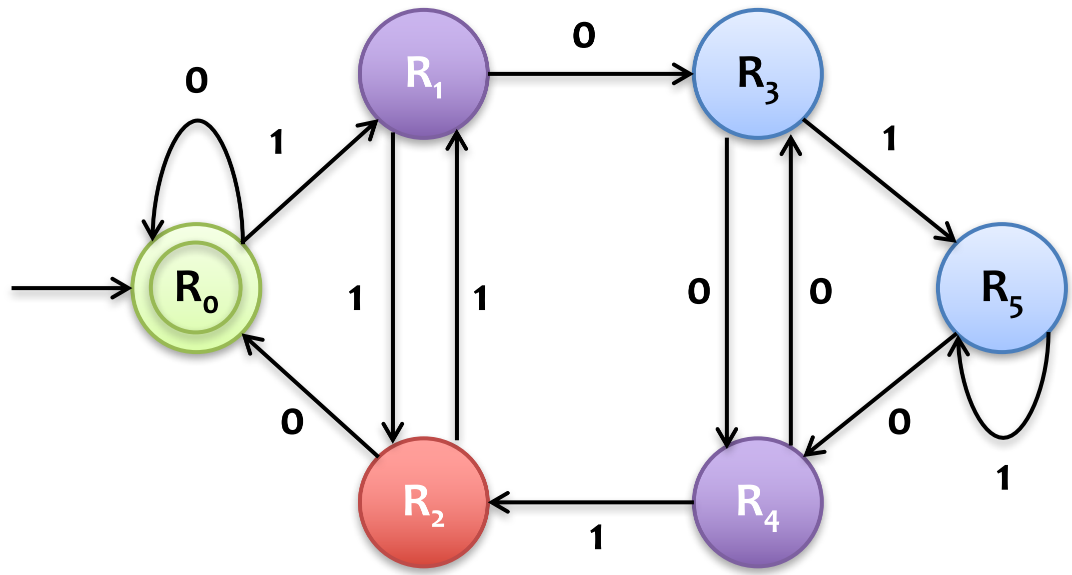
Grupos

$G_1 = \{R_0\}$

$G_2 = \{R_3, R_5\}$

$G_3 = \{R_2\}$

$G_4 = \{R_1, R_4\}$



Tem mais partições?

Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

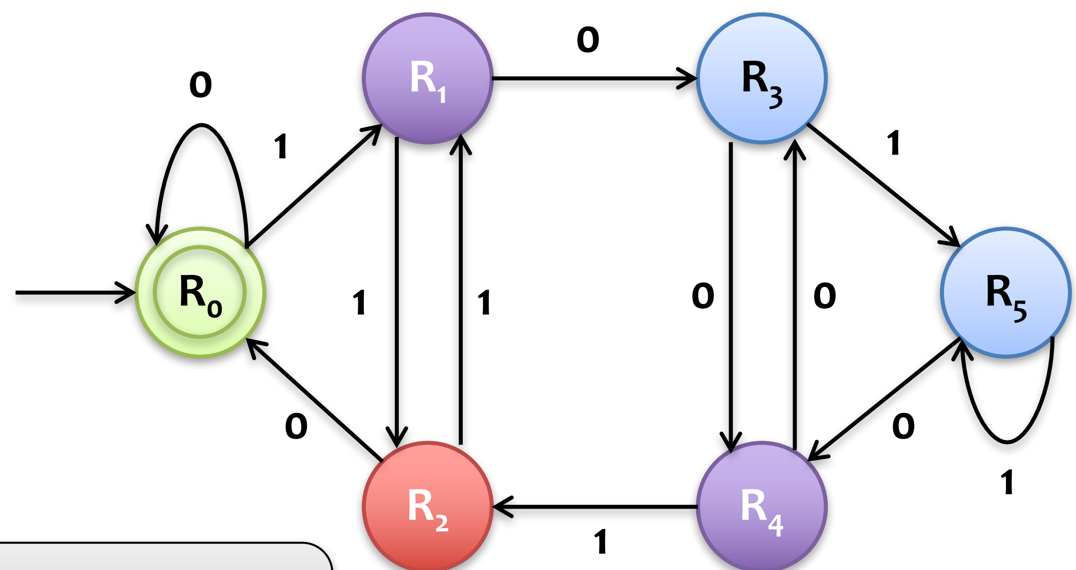
$G_1 = \{R_0\}$

$G_2 = \{R_3, R_5\}$

$G_3 = \{R_2\}$

$G_4 = \{R_1, R_4\}$

		0	1
G_1	R_0	G_1	G_4
G_4	R_1	G_2	G_3
G_3	R_2	G_1	G_4
G_2	R_3	G_4	G_2
G_4	R_4	G_2	G_3
G_2	R_5	G_4	G_2



Como gerar o AFD reduzido?

Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

$G_1 = \{R_0\}$

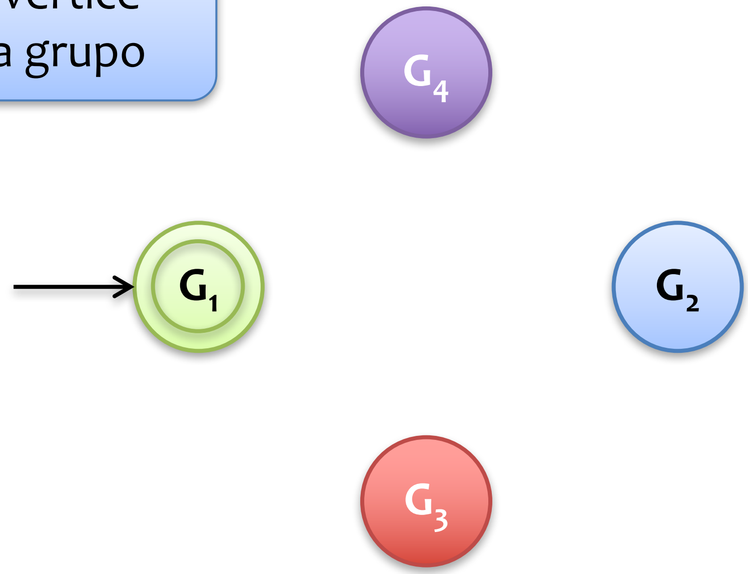
$G_2 = \{R_3, R_5\}$

$G_3 = \{R_2\}$

$G_4 = \{R_1, R_4\}$

		0	1
G_1	R_0	G_1	G_4
G_4	R_1	G_2	G_3
G_3	R_2	G_1	G_4
G_2	R_3	G_4	G_2
G_4	R_4	G_2	G_3
G_2	R_5	G_4	G_2

Criar um vértice para cada grupo



Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

Grupos

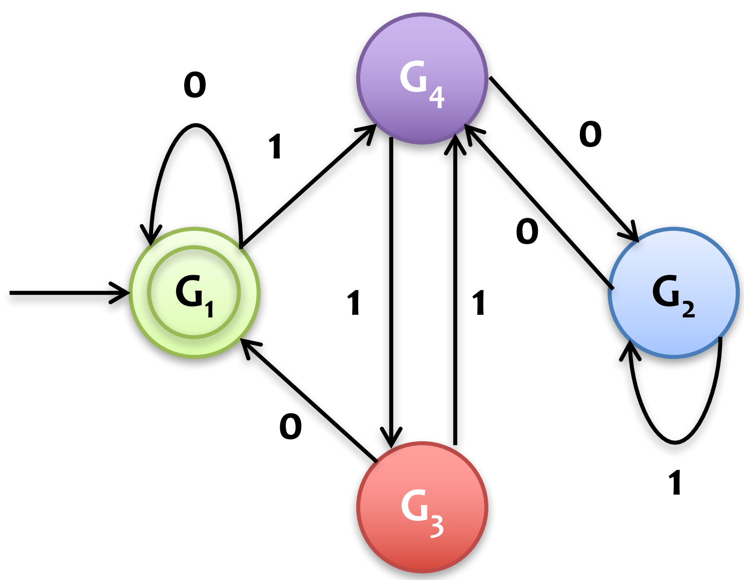
$G_1 = \{R_0\}$

$G_2 = \{R_3, R_5\}$

$G_3 = \{R_2\}$

$G_4 = \{R_1, R_4\}$

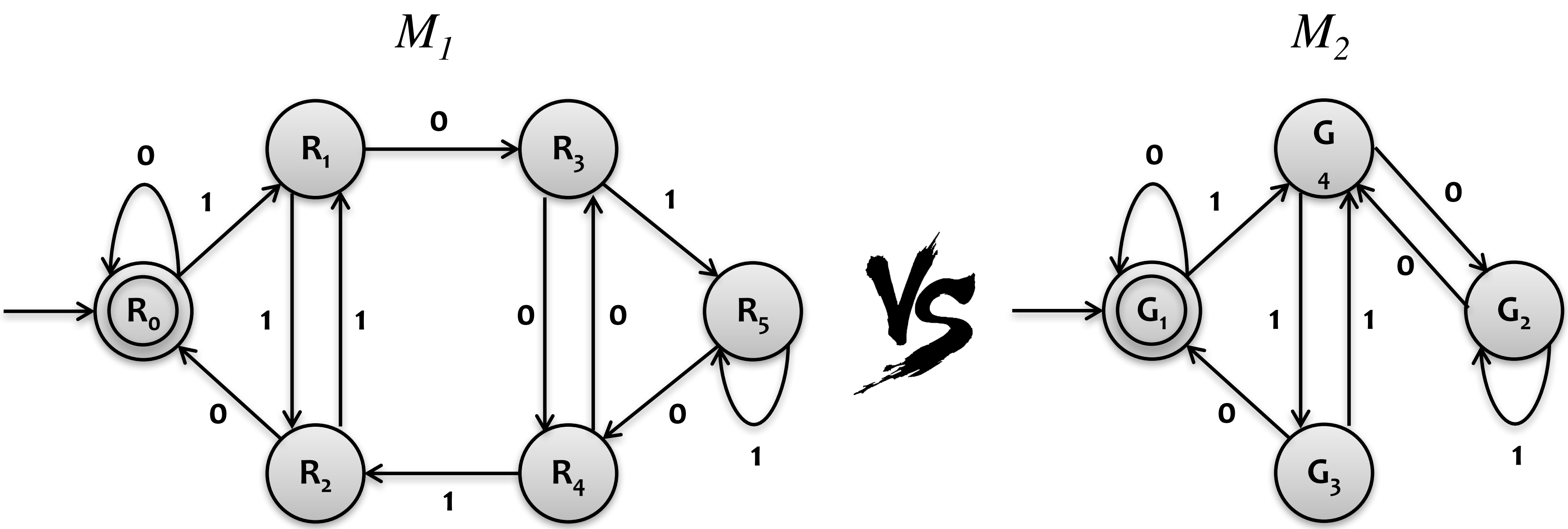
		0	1
G_1	R_0	G_1	G_4
G_4	R_1	G_2	G_3
G_3	R_2	G_1	G_4
G_2	R_3	G_4	G_2
G_4	R_4	G_2	G_3
G_2	R_5	G_4	G_2



Colocar as arestas baseado na tabela

Exemplo

- Para exemplificar o algoritmo de minimização, será utilizado o exemplo do problema da matemática (divisão por 6)

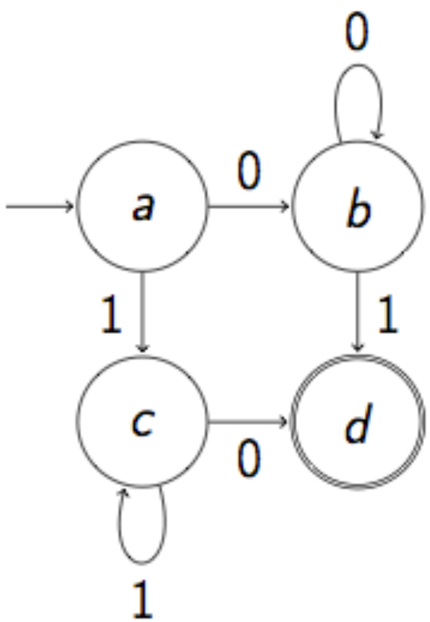


$$L(M_1) = L(M_2)$$

Exercícios

- Encontre o AFD reduzido para os autômatos a seguir

a)



b)