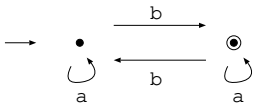
Autômatos e Linguagens Formais

aula 12: O autômato mínimo

1 Introdução

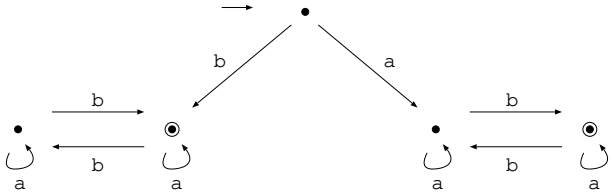
Às vezes um autômato pode parecer mais complicado do que ele realmente é ...

Por exemplo, o autômato abaixo aceita palavras com uma quantidade ímpar de `b`'s.



Esse autômato é simples e parece simples, não é?

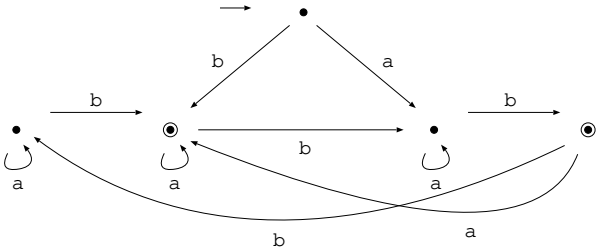
Agora, considere esse outro autômato que também aceita palavras com uma quantidade ímpar de `b`'s



Esse autômato já não é tão simples.

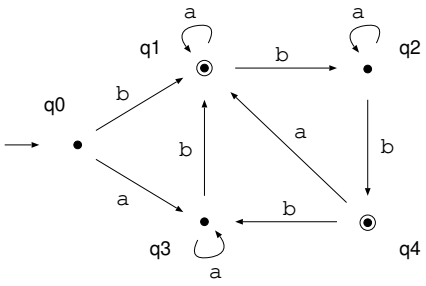
Mas ele também não é tão complicado assim.

A seguir, nós fazemos uma pequena mudança nas transições do autômato



mas isso não afeta as palavras que ele aceita — (*porque?*)

Finalmente, nós reorganizamos o diagrama da seguinte maneira

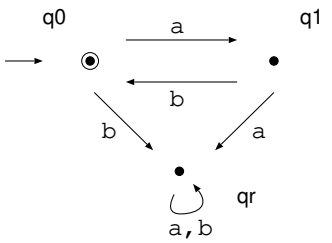


E nesse ponto, alguém que vê o autômato pela primeira vez pode perguntar: *O que será que esse autômato faz ?*

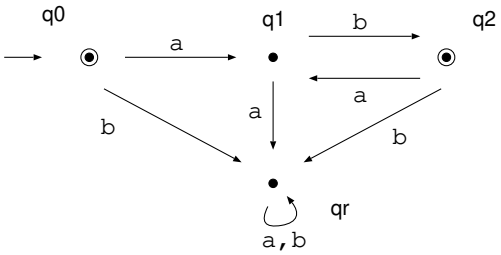
2 Complexidade aparente

Vejamos mais um exemplo.

O autômato abaixo reconhece as palavras da forma $(\mathtt{ab})^*$



E esse outro também

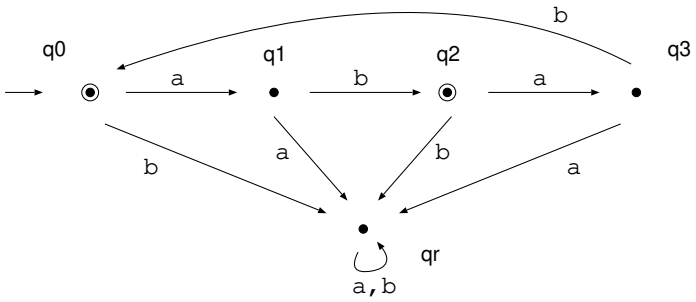


Você consegue ver como é que nós chegamos a esse segundo autômato?

Bom, a ideia é que o estado $\mathtt{q}_0$ foi duplicado, o que fez aparecer o estado $\mathtt{q}_2$.

E, como $\mathtt{q}_0$ e $\mathtt{q}_2$ são cópias um do outro, a transição de $\mathtt{q}_1$ associada ao símbolo $\mathtt{b}$ foi direcionada para o estado $\mathtt{q}_2$ (sem alterar a linguagem reconhecida pelo autômato).

Daí, a gente pode fazer a mesma coisa outra vez, duplicando o estado $\mathtt{q}_1$



Nesse ponto, alguém que vê o autômato pela primeira vez pode ficar imaginando o que é que ele faz.

E pode se perguntar se não existe um autômato mais simples que faz a mesma coisa.

Você sabe que sim.

Mas, você consegue ver como é que a gente simplifica um autômato?

Quer dizer, só olhando para o autômato, sem saber como ele foi construído?

3 Estados idênticos

A intuição que está surgindo aqui é que um autômato é mais complicado do que devia quando existe redundância lógica.

Quer dizer, quando uma parte do autômato faz a mesma coisa que outra parte — (*elas tem a mesma função*)

O caso mais simples de redundância corresponde aos *estados idênticos*.

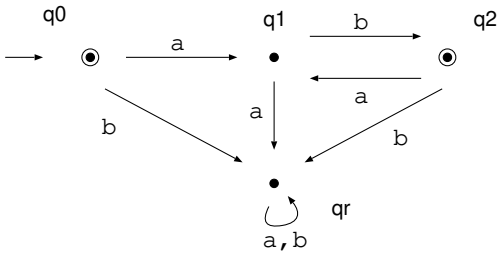
Quer dizer, nós dizemos que dois estados q_i e q_j são idênticos quando

- eles tem o mesmo tipo — (*i.e., final ou não-final*)
- e as suas transições levam para os mesmos lugares

Intuitivamente, não faz a menor diferença se o autômato está em q_i ou q_j em algum ponto da computação — (*porque?*)

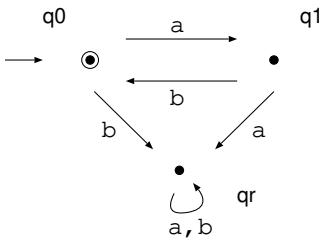
Daí que, nesse tipo de situação, nós sempre podemos eliminar um deles, e redirecionar as transições que chegam nele para o outro estado.

Por exemplo, considere outra vez o autômato



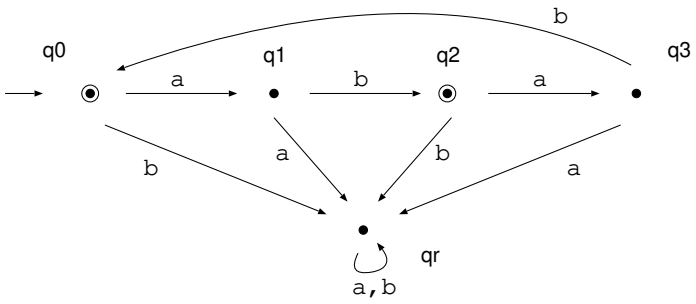
Aqui, não é difícil ver que q_0 e q_2 são estados idênticos.

E daí, eliminando o estado q_2, nós obtemos



que é a versão mais simples do autômato que reconhece as palavras da forma $(\mathtt{ab})^*$.

Mas, começando com o autômato

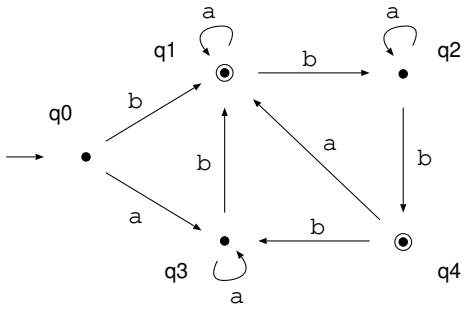


nós não conseguimos identificar nenhum par de estados idênticos.

E não conseguimos reduzir o autômato à sua versão mais simples.

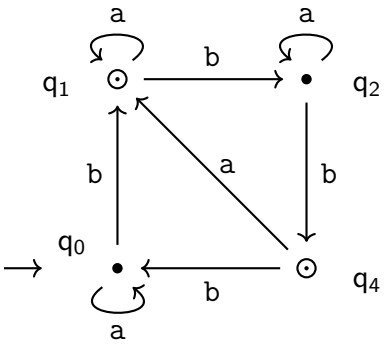
Vejamos mais um exemplo.

Considere outra vez o autômato



Aqui, nós vemos que q_0 e q_3 são estados idênticos.

E isso nos permite simplificar o autômato para



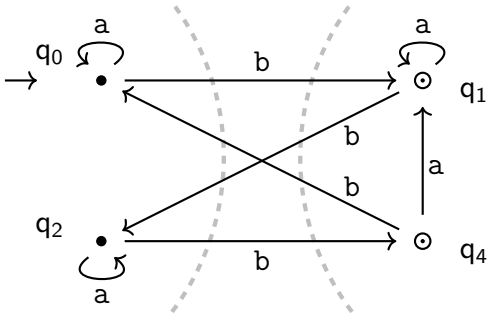
Mas agora, já não há mais como continuar ...

4 Estados equivalentes

Certo.

Agora nós precisamos de uma ideia nova.

E a intuição aparece quando nós reorganizamos o autômato acima da seguinte maneira



Quer dizer, tudo o que importa é se a gente está no lado esquerdo ou no lado direito do diagrama.

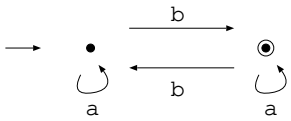
Porque no lado esquerdo a palavra é rejeitada, e no lado direito ela é aceita.

Além disso,

- o símbolo `a` sempre mantém a gente do mesmo lado
- e o símbolo `b` faz a gente mudar de lado

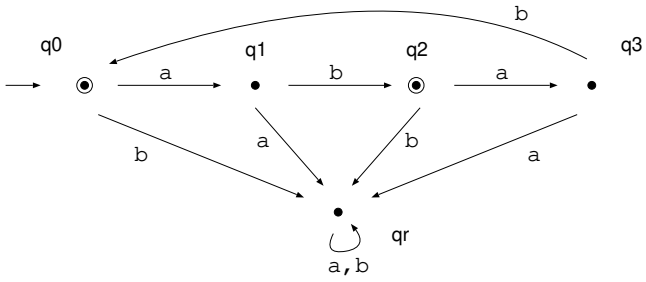
Quer dizer, a intuição é que os dois estados de cada lado são *equivalentes*
— (*não importa em qual deles a gente está ...*)

E com base nessa intuição, a gente reconstrói o autômato assim

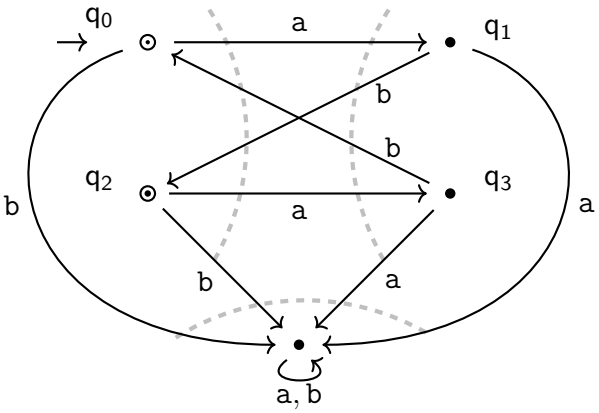


Legal.

Agora considere outra vez o autômato



A ideia aqui é reorganizar o autômato assim

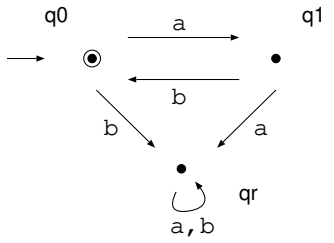


Quer dizer, dessa vez tudo o que importa é se a gente está

- no lado esquerdo
- no lado direito
- no estado de rejeição

Além disso, as transições com o mesmo símbolo que começam no mesmo lugar, sempre levam a gente para o mesmo lugar.

Daí que, a gente só precisa de 3 estados para poder representar essa lógica



Legal, não é?

5 Método sistemático

Certo.

Então a ideia é descobrir os grupos de estados equivalentes do autômato.

E daí, a gente substitui cada grupo por um único estado.

Mas, como é que a gente faz isso?

Bom, na prática é mais fácil descobrir quando dois estados não são equivalentes
— (*e devem estar em grupos diferentes*)

Por exemplo, se q_i é um estado final e q_j é um estado não final

então eles certamente não são equivalentes — (*estar em* q_i *e estar em* q_j *não é a mesma coisa*)

Da mesma forma, se q_i leva para um estado final e q_j leva para um estado não final, com o símbolo a

então q_i e q_j também não são equivalentes.

Porque não é a mesma coisa estar em q_i e estar em q_j.

Quer dizer, se falta apenas um a para acabar a palavra de entrada, então estar em q_i levaria para a resposta SIM, e estar em q_j levaria para a resposta NÃO.

Pronto.

É só isso.

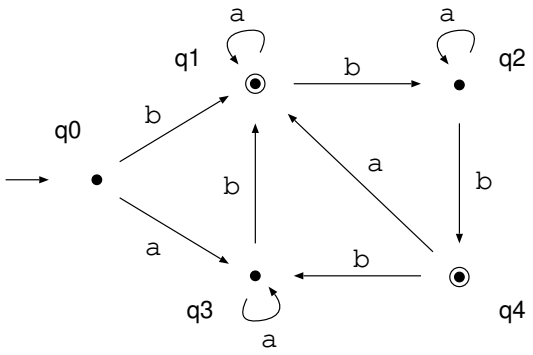
Quer dizer, a regra é a seguinte

- Se q_i e q_j possuem tipos diferentes, então eles não são equivalentes
  — (*e devem estar em grupos diferentes*)
- Se q_i e q_j levam para grupos diferentes, com o símbolo `a` ou `b`,
  então eles não são equivalentes — (*e devem estar em grupos diferentes*)

Agora, basta aplicar essas regras em um processo repetitivo para simplificar o autômato.

Vejamos um exemplo concreto.

Considere o seguinte autômato outra vez



No início, a única coisa que a gente sabe é que q_1, q_4 são estados finais, e que q_0, q_2, q_3 são estados não finais

$$F = \{ q_1, q_4 \} \qquad NF = \{ q_0, q_2, q_3 \}$$

Daí, a gente examina, dentro de cada grupo, para onde as coisas vão

(F)	a	b
q_1	F	NF
q_4	F	NF

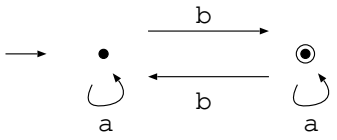
(NF)	a	b
q_0	NF	F
q_2	NF	F
q_3	NF	F

Dentro de cada grupo, os estados tem o mesmo comportamento.

Logo a coisa acabou.

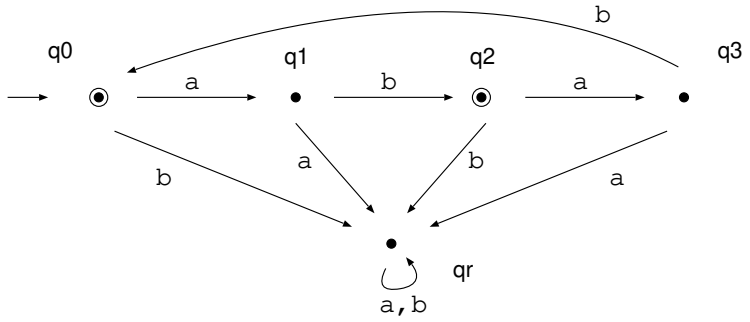
Quer dizer, dentro de cada grupo os estados são todos equivalentes.

Daí, nós podemos simplificar o autômato da seguinte maneira



Vejamos outro exemplo.

Considere o seguinte autômato outra vez



No início, só o que nos sabemos é

$$F = \{ q_0, q_2 \} \qquad NF = \{ q_1, q_3, q_r \}$$

Daí, nós examinamos o comportamento dos estados dentro de cada grupo

(F)	a	b
q_0	NF	NF
q_2	NF	NF

(NF)	a	b
q_1	NF	F
q_3	NF	F
q_r	NF	NF

Quer dizer, os estados em NF não possuem todos o mesmo comportamento.

Daí, nós subdividimos o grupo NF e obtemos o seguinte agrupamento

$$A_1 = \{ q_0, q_2 \} \qquad A_2 = \{ q_1, q_3 \} \qquad A_3 = \{ q_r \}$$

E examinamos outra vez o comportamento dos estados dentro de cada grupo

(A_1)	a	b
q_0	A_2	A_3
q_2	A_2	A_3

(A_2)	a	b
q_1	A_3	A_1
q_3	A_3	A_1

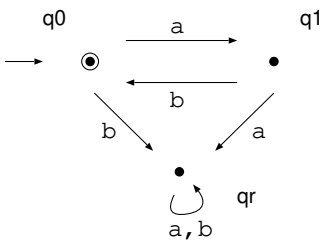
(A_3)	a	b
q_r	A_3	A_3

Pronto.

Dentro de cada grupo, os estados possuem o mesmo comportamento.

Logo, eles são todos equivalentes.

E daí, nós podemos simplificar o autômato assim



Não é legal?

6 O autômato mínimo

Sim, é legal.

Mas quem garante que esse método produz o menor autômato possível ?

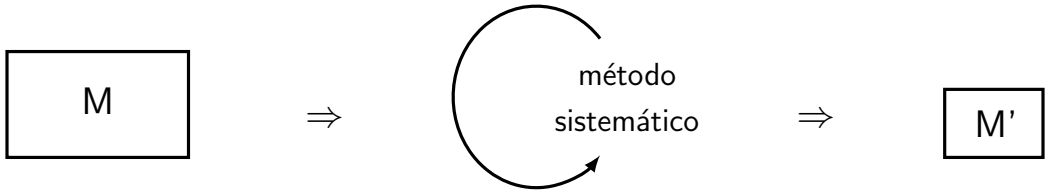
— (que aceita o mesmo conjunto de palavras)

Hmm ...

Agora a gente precisa raciocinar um pouquinho.

Quer dizer, imagine que a gente tinha um autômato M.

Imagine que a gente aplicou o método sistemático da Seção 5 para obter o autômato M′



E imagine que M′ tem n estados.

Agora suponha que M* é um autômato qualquer com menos do que n estados.

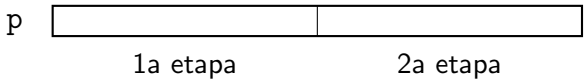
Bom, a ideia é que M* não pode aceitar o mesmo conjunto de palavras que o autômato M′.

E para demonstrar isso, nós vamos encontrar uma palavra p que M′ aceita e M* rejeita, ou vice-versa.

Certo.

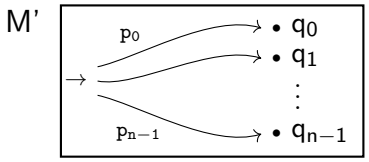
Nós vamos fazer o raciocínio em 2 etapas.

E a ideia é que cada etapa vai encontrar uma parte da palavra p



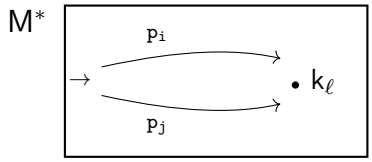
A primeira etapa coloca em foco os k estados do autômato M′

E daí, nós encontramos n palavras $p_0, p_1, \ldots, p_{n-1}$ que levam o autômato M′ do estado inicial a cada um desses estados



O próximo passo é fornecer as palavras $\mathtt{p}_0, \mathtt{p}_1, \ldots, \mathtt{p}_{n-1}$ como entrada para o autômato M^*.

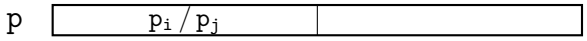
E daí, como M^* tem menos do que k estados, certamente existem palavras $\mathtt{p}_i$ e $\mathtt{p}_j$ que levam M^* para o mesmo estado



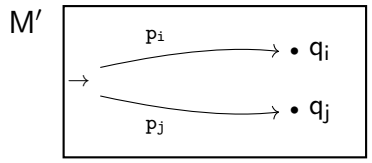
Pronto.

A primeira etapa acaba aqui.

E $\mathtt{p}_i$ ou $\mathtt{p}_j$ vão nos dar a primeira parte da palavra $\mathtt{p}$

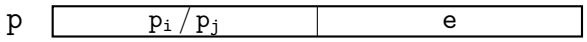


A segunda etapa coloca em foco os estados $\mathtt{q}_i$ e $\mathtt{q}_j$ do autômato M', que foram alcançados pelas palavras $\mathtt{p}_i$ e $\mathtt{p}_j$



E o caso mais simples acontece quando $\mathtt{q}_i$ é um estado final, e $\mathtt{q}_j$ é um estado não final
— (*ou vice-versa*)

Nesse caso, a segunda parte da palavra $\mathtt{p}$ é a palavra vazia $\mathtt{e}$



Quer dizer, imagine que o autômato M' recebe a palavra $(\mathtt{p}_i\,\mathtt{e})$ como entrada.

Nesse caso, a computação termina no estado $\mathtt{q}_i$, e o autômato responde SIM.

Agora imagine que o autômato M' recebe a palavra $(\mathtt{p}_j\,\mathtt{e})$ como entrada.

Nesse caso, a computação termina no estado $\mathtt{q}_j$, e o autômato responde NÃO.

O ponto importante aqui é que o autômato M' dá respostas diferentes para as palavras $(\mathtt{p}_i\,\mathtt{e})$ e $(\mathtt{p}_j\,\mathtt{e})$.

Agora imagine que as palavras $(\mathtt{p}_i\,\mathtt{e})$ e $(\mathtt{p}_j\,\mathtt{e})$ são fornecidas como entrada para o autômato M^*.

Então, nos dois casos a computação termina no estado $\mathtt{k}_\ell$.

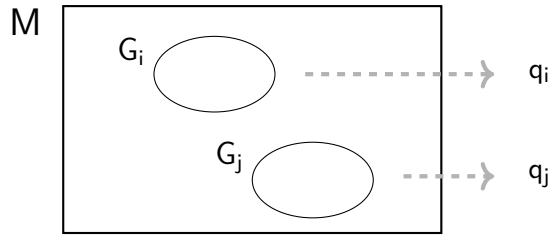
E isso significa que o autômato M^* dá a mesma resposta para as palavras $(\mathtt{p}_i\,\mathtt{e})$ e $(\mathtt{p}_j\,\mathtt{e})$.

Portanto, o autômato M^* não aceita exatamente o mesmo conjunto de palavras que o autômato M'.

Legal.

Agora imagine que q_i e q_j são estado do mesmo tipo — (*i.e., ambos finais ou ambos não-finais*)

Daí, a gente observa que q_i e q_j correspondem a dois grupos diferentes de estados do autômato original M, calculados pelo método sistemático



E nesse ponto, a gente pergunta como é que os grupos G_i e G_j vieram a se separar um do outro.

Bom, a separação pode ter ocorrido logo no primeiro passo do método, quando a gente descobriu (por exemplo) que

$$G_i \xrightarrow{a} F \qquad\qquad G_j \xrightarrow{a} NF$$

Ou então, a separação pode ter acontecido no segundo passo do método, quando a gente descobriu (por exemplo) que

$$G_i \xrightarrow{a} A_1 \xrightarrow{b} F \qquad\qquad G_j \xrightarrow{a} A_3 \xrightarrow{b} NF$$

Ou então, a separação pode ter acontecido no terceiro passo do método, quando a gente descobriu (por exemplo) que

$$G_i \xrightarrow{a} A_1 \xrightarrow{b} B_2 \xrightarrow{a} F \qquad\qquad G_j \xrightarrow{a} A_3 \xrightarrow{b} B_1 \xrightarrow{a} F$$

E por aí vai ...

Quer dizer, quando os grupos G_i e G_j são separados pelo método sistemático, nós sempre podemos encontrar uma palavra x que

- leva os estados do grupo G_i para F
- e leva os estados do grupo G_j para NF

— (*ou vice-versa*)

Essa palavra x é a segunda parte da palavra p que nós estamos montando.

E agora basta repetir o argumento que foi apresentado acima.

7 Unicidade do autômato mínimo

Suponha que M_1 e M_2 são dois autômatos mínimos que reconhecem a linguagem L.

Na aula 14, nós vimos que $\mathsf{M_1}$ e $\mathsf{M_2}$ devem ter o mesmo número de estados.

Mas, será que eles são o mesmo autômato?

Ou será que eles podem ser autômatos diferentes?

Vejamos.

De acordo com a definição, dois estados $\mathsf{q_i}$ e $\mathsf{q_j}$ de um autômato M são equivalentes se, para toda palavra $\mathtt{p}$, nós temos que

- $\mathsf{q_i} \xrightarrow{\mathtt{p}}_M \bullet$ e $\mathsf{q_j} \xrightarrow{\mathtt{p}}_M \bullet$

ou

- $\mathsf{q_i} \xrightarrow{\mathtt{p}}_M \odot$ e $\mathsf{q_j} \xrightarrow{\mathtt{p}}_M \odot$

Isto é, o resultado de uma computação que começa em $\mathsf{q_i}$ e uma computação que começa em $\mathsf{q_j}$ é sempre o mesmo, para qualquer palavra $\mathtt{p}$.

Mas nós também podemos usar essa definição para estados em autômatos diferentes.

Suponha que $\mathsf{q_0}$ é o estado inicial do autômato $\mathsf{M_1}$ e $\mathsf{k_0}$ é o estado inicial do autômato $\mathsf{M_2}$.

Então, $\mathsf{q_0}$ e $\mathsf{k_0}$ certamente são equivalentes.

Porque se a palavra $\mathtt{p}$ pertence à linguagem L, então

- $\mathsf{q_0} \xrightarrow{\mathtt{p}}_{M_1} \odot$ e $\mathsf{k_0} \xrightarrow{\mathtt{p}}_{M_2} \odot$

E se a palavra $\mathtt{p}$ não pertence à linguagem L, então

- $\mathsf{q_0} \xrightarrow{\mathtt{p}}_{M_1} \bullet$ e $\mathsf{k_0} \xrightarrow{\mathtt{p}}_{M_2} \bullet$

Além disso, $\mathsf{k_0}$ não pode ser equivalente a nenhum outro estado de $\mathsf{M_1}$, pois os outros estados de $\mathsf{M_1}$ não são equivalentes a $\mathsf{q_0}$.

Da mesma forma, $\mathsf{q_0}$ não pode ser equivalente a nenhum outro estado de $\mathsf{M_2}$.

Certo.

Agora, suponha que a leitura do símbolo $\mathtt{a}$ em $\mathsf{M_1}$ a partir do estado $\mathsf{q_0}$ leva o autômato para o estado $\mathsf{q_1}$.

E que a leitura do símbolo $\mathtt{a}$ em $\mathsf{M_2}$ a partir do estado $\mathsf{k_0}$ leva o autômato para o estado $\mathsf{k_1}$.

Então, os estados $\mathsf{q_1}$ e $\mathsf{k_1}$ também são equivalentes.

Vejamos porque.

Suponha que $\mathtt{p}$ é uma palavra qualquer, e considere a palavra $\mathtt{p}' = \mathtt{ap}$.

Então, se $\mathtt{p}'$ pertence à linguagem L, nós temos que

- $\mathsf{q_0} \xrightarrow{\mathtt{a}} \mathsf{q_1} \xrightarrow{\mathtt{p}}_{M_1} \odot$ e $\mathsf{k_0} \xrightarrow{\mathtt{a}} \mathsf{k_1} \xrightarrow{\mathtt{p}}_{M_2} \odot$

E se $\mathtt{p}'$ não pertence à linguagem L, nós temos que

- $\mathsf{q}_0 \xrightarrow{\mathtt{a}} \mathsf{q}_1 \overset{\mathtt{p}}{\rightsquigarrow}_{M_1} \bullet$ e $\mathsf{k}_0 \xrightarrow{\mathtt{a}} \mathsf{k}_1 \overset{\mathtt{p}}{\rightsquigarrow}_{M_2} \bullet$

Além disso, k_1 não pode ser equivalente a nenhum outro estado de M_1.

E q_1 não pode ser equivalente a nenhum outro estado de M_2.

Legal.

Raciocinando dessa maneira, é possível associar os estados de M_1 e M_2 (de maneira única) em pares $(\mathsf{q}_i, \mathsf{k}_i)$ onde q_i e k_i são estados equivalentes.

Não apenas isso, mas também

- $\mathsf{q}_i \xrightarrow{\mathtt{a}} \mathsf{q}_j$ sse $\mathsf{k}_i \xrightarrow{\mathtt{a}} \mathsf{k}_j$

e

- $\mathsf{q}_i \xrightarrow{\mathtt{b}} \mathsf{q}_j$ sse $\mathsf{k}_i \xrightarrow{\mathtt{b}} \mathsf{k}_j$

Ou seja, os autômatos M_1 e M_2 possuem basicamente os mesmos estados e o mesmo diagrama de transições.

Logo, M_1 e M_2 são o mesmo autômato! :)