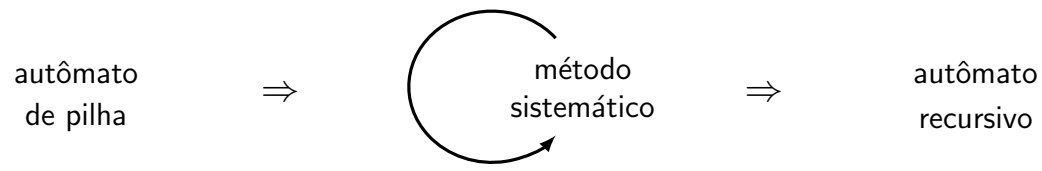
## Método sistemático

E agora é só repetir




Mas antes, é preciso resolver dois pequenos problemas.

O primeiro deles é que em geral as transições fazem mais de uma coisa

$$q_i \xrightarrow{a[b,c]} q_j$$

Mas isso é fácil de resolver, decompondo a transição assim

$$q_i \xrightarrow{e[b,e]} \bullet \xrightarrow{a} \bullet \xrightarrow{e[e,c]} q_j$$

Quer dizer, primeiro acontece o retorno de alguma função, depois a leitura do símbolo na palavra de entrada, e depois a chamada de uma função — (*mas qualquer outra ordem também funciona — porque?*)

O caso de operações na pilha com múltiplos símbolos também não apresenta maiores problemas, pois nós sempre podemos decompor a transição

$$q_i \xrightarrow{a[bb,cc]} q_j$$

como um caminho

$$q_i \xrightarrow{e[b,e]} \bullet \xrightarrow{e[b,e]} \bullet \xrightarrow{a} \bullet \xrightarrow{e[e,c]} \bullet \xrightarrow{e[e,c]} q_j$$

O segundo problema é o seguinte.

Quando nós fizemos o raciocínio na seção anterior, nós estávamos trabalhando com uma *linha de execução* específica do autômato.

Mas, a conversão precisa funcionar para todas as execuções possíveis.

Em particular, o símbolo que é colocado na pilha pela transição

$$q_i \xrightarrow{e[e,a]} q_j$$

pode ser removido da pilha por transições diferentes em execuções diferentes do autômato

$$q_k \xrightarrow{e[a,e]} q_l \qquad q_m \xrightarrow{e[a,e]} q_n \qquad q_u \xrightarrow{e[a,e]} q_v$$

Quer dizer, para interpretar a transição

$$q_i \xrightarrow{e[e,a]} q_j$$

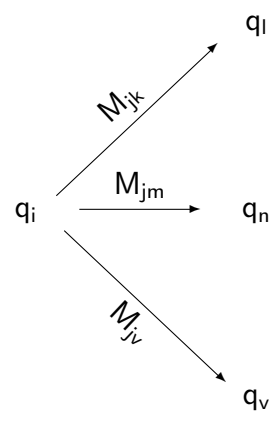
como uma chamada de função, é preciso saber já no momento da chamada qual é o lugar de retorno (ou o estado que vai remover o símbolo da pilha), para saber qual é o módulo que está sendo chamado.

No exemplo acima, nós teríamos 3 possibilidades

$$q_i \xrightarrow{M_{jk}} q_l \qquad q_i \xrightarrow{M_{jm}} q_n \qquad q_i \xrightarrow{M_{ju}} q_v$$

Mas, nós podemos deixar essa questão ser resolvida pelo não-determinismo (em tempo de execução).

Quer dizer, nós podemos substituir as transições que manipulam a pilha pela coleção de chamadas de função possíveis




Pronto.

Esses eram os problemas que precisavam ser resolvidos.