Autômatos recursivos

Lembre que o procedimento de composição recursiva consiste em introduzir um bloco no interior de um outro bloco.

Por exemplo,

```
a . . . . b
  a b
```

Logo, para reconhecer esse tipo de coisa, nós precisamos ter a capacidade de

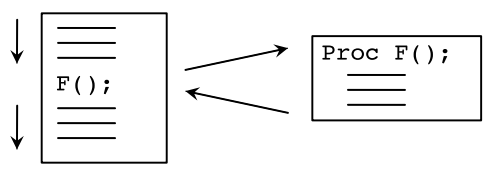
- iniciar a leitura de um bloco no meio da leitura de outro bloco
- e depois voltar para terminar a leitura do bloco que foi interrompida

De maneira mais geral, isso corresponde à capacidade de

- iniciar uma computação no meio de uma outra computação
- e depois voltar para terminar a computação que foi interrompida

E isso corresponde a uma *chamada de função*.

Quer dizer, quando o nosso programa faz uma chamada de função



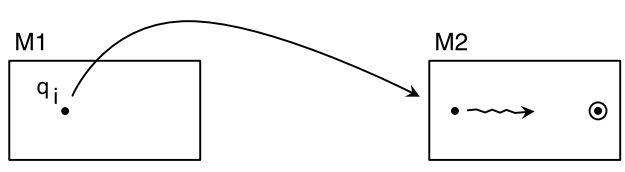
o fluxo de execução que vinha sendo realizado é interrompido por um momento, daí o código da função é executado e, quando ele termina, o fluxo de execução interrompido é retomado outra vez.

Mas, como é que isso funciona no mundo dos autômatos?

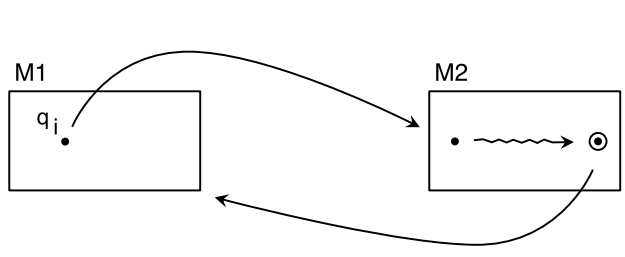
Bom, nós já vimos uma ideia semelhante na aula 02, quando nós construímos autômatos que tinham outros autômatos como componentes.

Quer dizer, a ideia é que nós podemos ter dois autômatos M_1 e M_2

e que, em um certo ponto da computação de M_1, ao invés do autômato ler um símbolo da palavra de entrada, ele dá início a uma computação de M_2



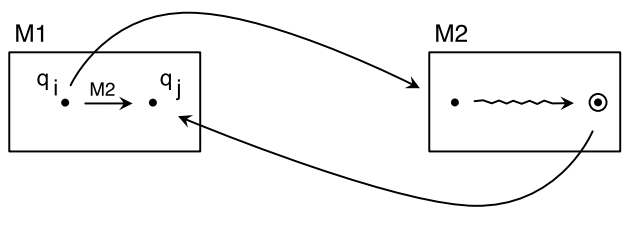
Daí, quando a computação de M_2 acaba, o controle volta para o autômato M_1



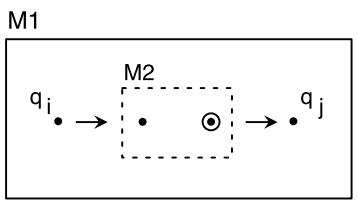
A maneira natural de operacionalizar essa ideia consiste em definir um novo tipo de transição que corresponde à ideia de chamada de função

$$q_i \xrightarrow{M_2} q_j$$

A ideia é que, no momento em que o controle volta retorna para o autômato M_1, a computação continua a partir do estado q_j



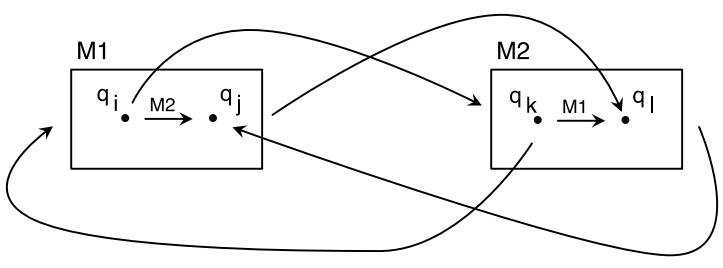
À primeira vista, isso não parece adicionar poder computacional aos autômatos, pois nós sempre podemos reorganizar a coisa da seguinte maneira



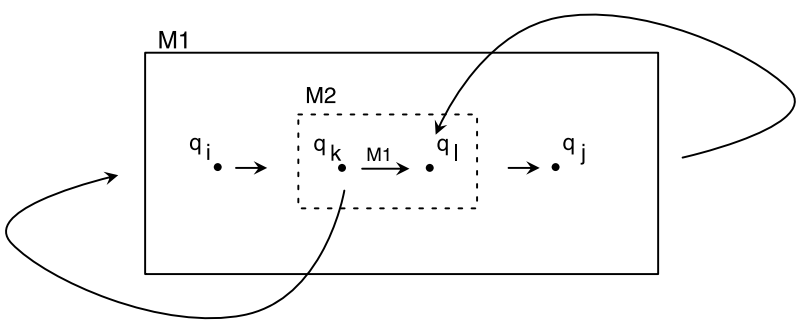
Quer dizer, nesse esquema, M_2 não é realmente um autômato separado, mas apenas uma parte do autômato M_1 que nós podemos querer desenhar em separado.

— (*Era exatamente isso o que estava acontecendo na aula 02*)

Mas, a coisa fica realmente interessante quando a gente se dá conta de que, em princípio, o autômato M_2 também pode fazer uma chamada ao autômato M_1



E daí, quando a gente embute o autômato M_2 dentro do autômato M_1 outra vez



o que a gente vê é o autômato M_1 fazendo uma chamada (recursiva) a si mesmo.

Essa é a chave para o reconhecimento de padrões recursivos.

E é isso que dá um ganho de poder computacional aos autômatos.