

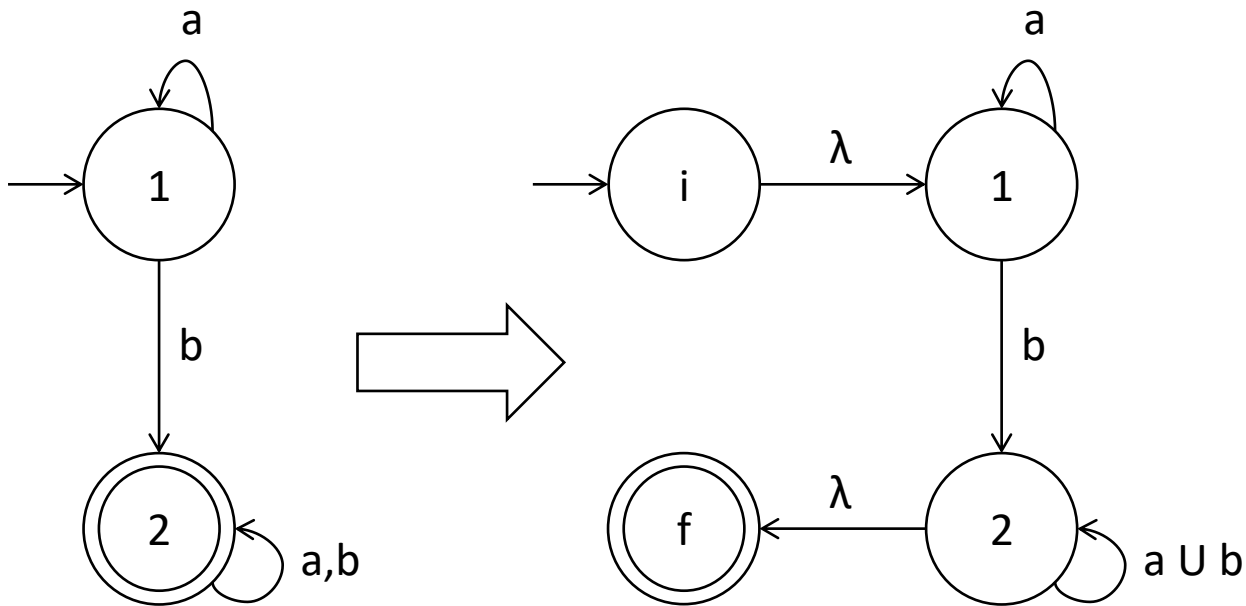
Equivalência entre ERs & AFNDs & Lema do bombeamento

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC
Professor: Andrey Brito Período: 2023.2

Equivalência AFND \rightarrow ER

- A transformação nem sempre é intuitiva, precisamos de um processo
- Mas em geral convertemos o autômato eliminando os estados um a um até que sobrem somente dois estados
 - Estes estados são um estado inicial e um final
 - O estado inicial tem uma única transição, saindo dele
 - O estado final tem uma única transição, chegando nele
 - Quando eliminarmos um estado, vamos compensar a remoção adicionando símbolos a outras transições diretas
 - As transições podem ter agora expressões regulares, que vão crescendo para representar partes cada vez maiores do autômatos

Preparando o AFND (intuitivamente)



- Adicione os novos estados inicial e final, **sem alterar a funcionalidade do AFND**
- Transforme as transições em ERs
- Remova estados, um a um, e para **cada remoção, adicione transições** (ou modifique as existentes para manter a funcionalidade)

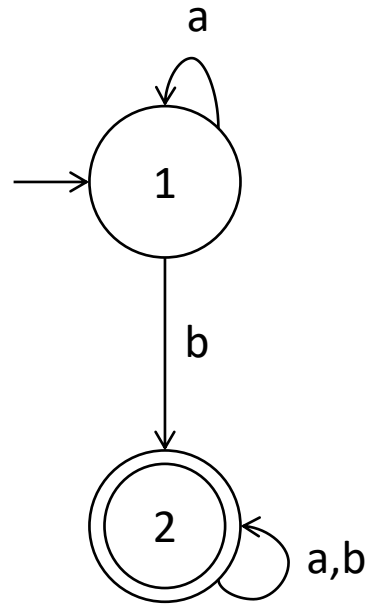
Autômato finito não-determinístico generalizado

- AFNDGs: AFNDs com arcos que são expressões regulares
 - Podem consumir vários símbolos de uma vez
 - Útil no processo de transformação
- Autômatos com um formato específico (**por quê?**)
 1. O estado inicial tem arcos levando a todos os estados, mas nenhum arco chegando nele
 2. O estado final tem arcos chegando de todos os estados, mas nenhum arco saindo dele
 3. Só existe um estado de aceitação, ele não pode ser o estado inicial
 4. Excluindo inicial e final, os estados têm um arco indo para todos os estados (inclusive para si mesmo)

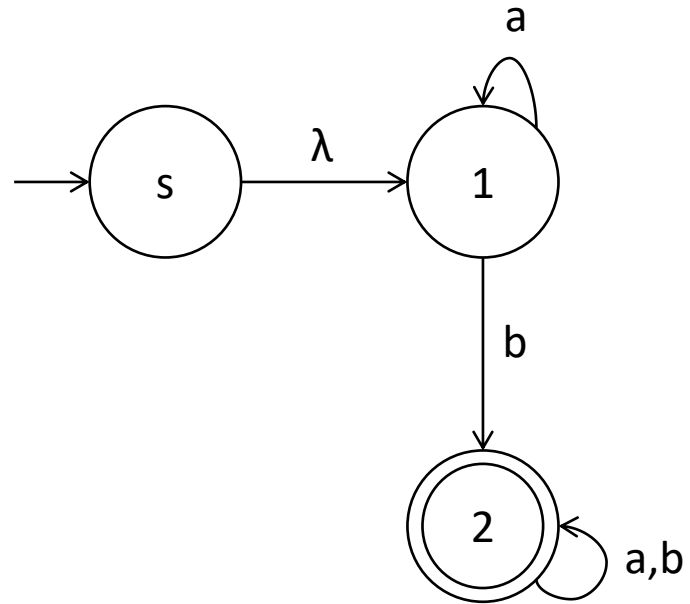
AFNDG

- Como converter um autômato para essa forma especial?
 - Adicione um novo estado inicial com arco λ para o antigo estado inicial
 - Adicione um novo estado final com arcos λ vindo dos antigos estados finais
 - Se qualquer arco tem vários rótulos, substitua-os pelo rótulo igual à união deles
 - Adicione arcos rotulados por \emptyset entre estados que não tem arcos

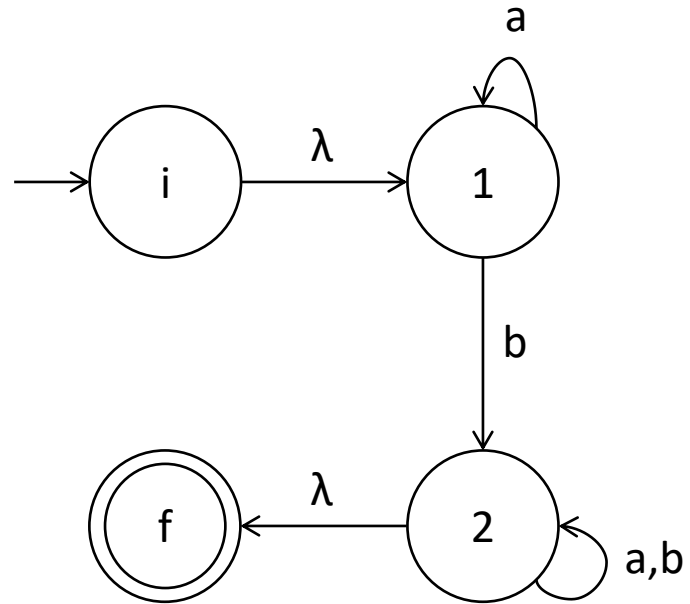
Gerando o AFNDG



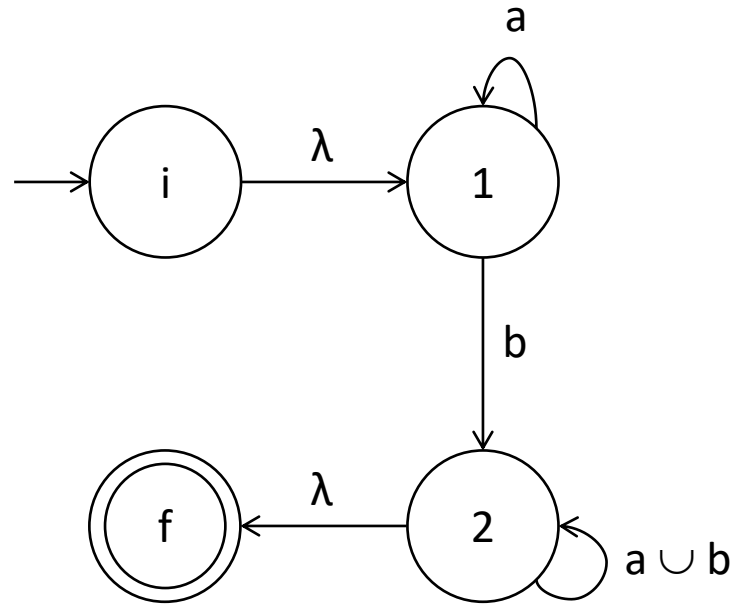
Gerando o AFNDG



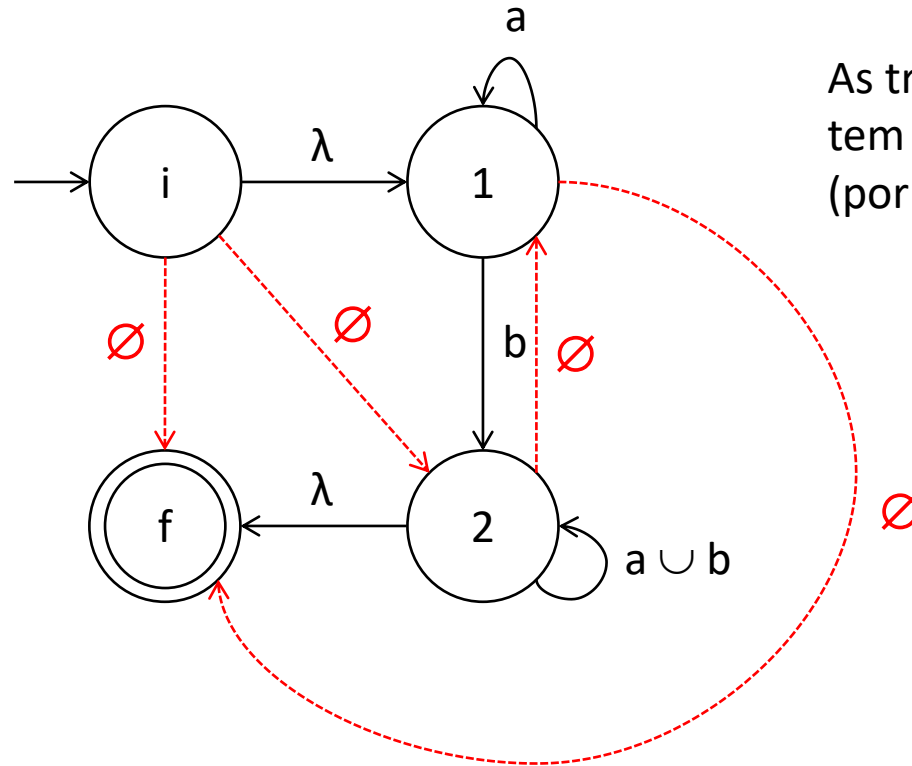
Gerando o AFNDG



Gerando o AFNDG



Gerando o AFNDG

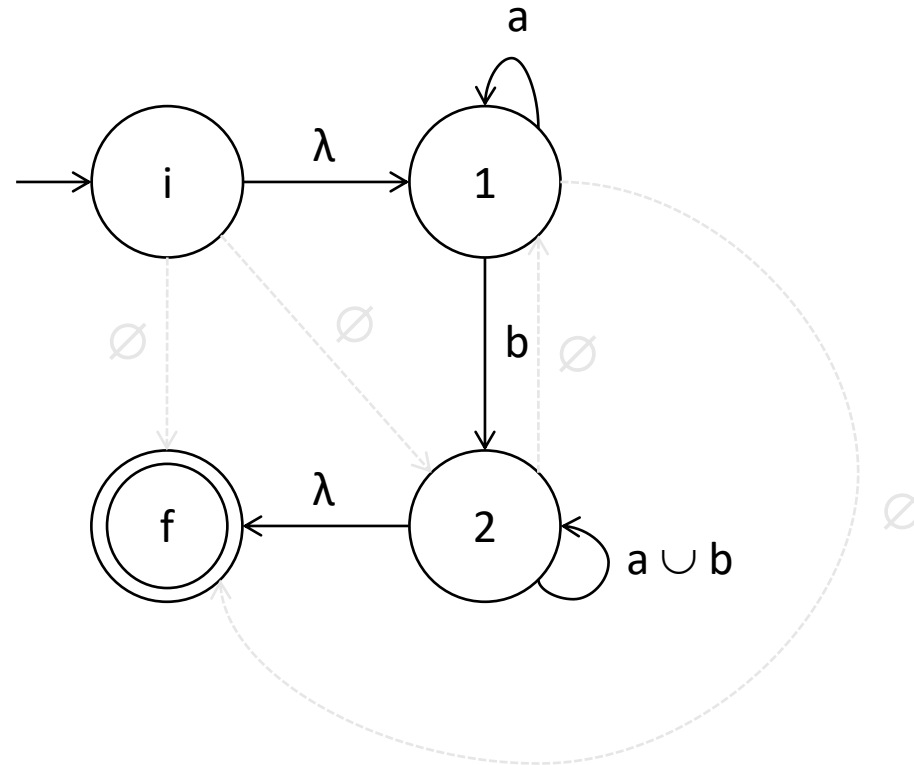


As transições de vermelho não tem nenhum efeito prático (por quê?).

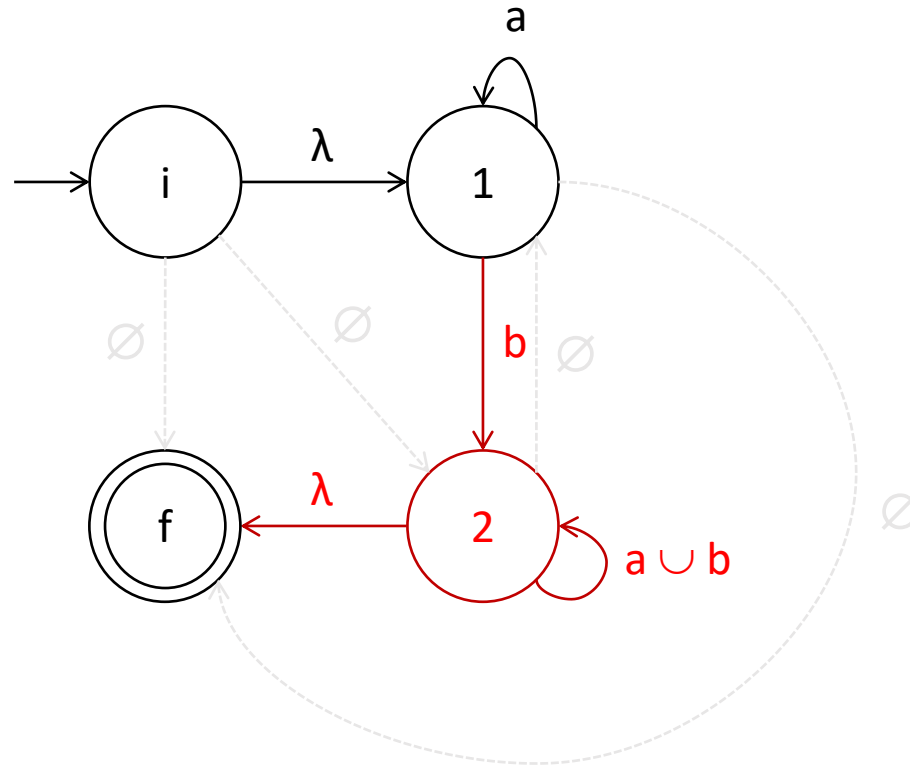
De AFNGs para expressões regulares

- Comece com um AFD com k estados
- Gere um AFNDG com $k+2$ estados (seguindo os passos do slide anterior)
- Gere um AFNDG com um estado a menos, sem perder a forma especial
 - Repita o processo até que o AFNDG tenha apenas 2 estados
 - O rótulo da transição restante é a expressão regular equivalente

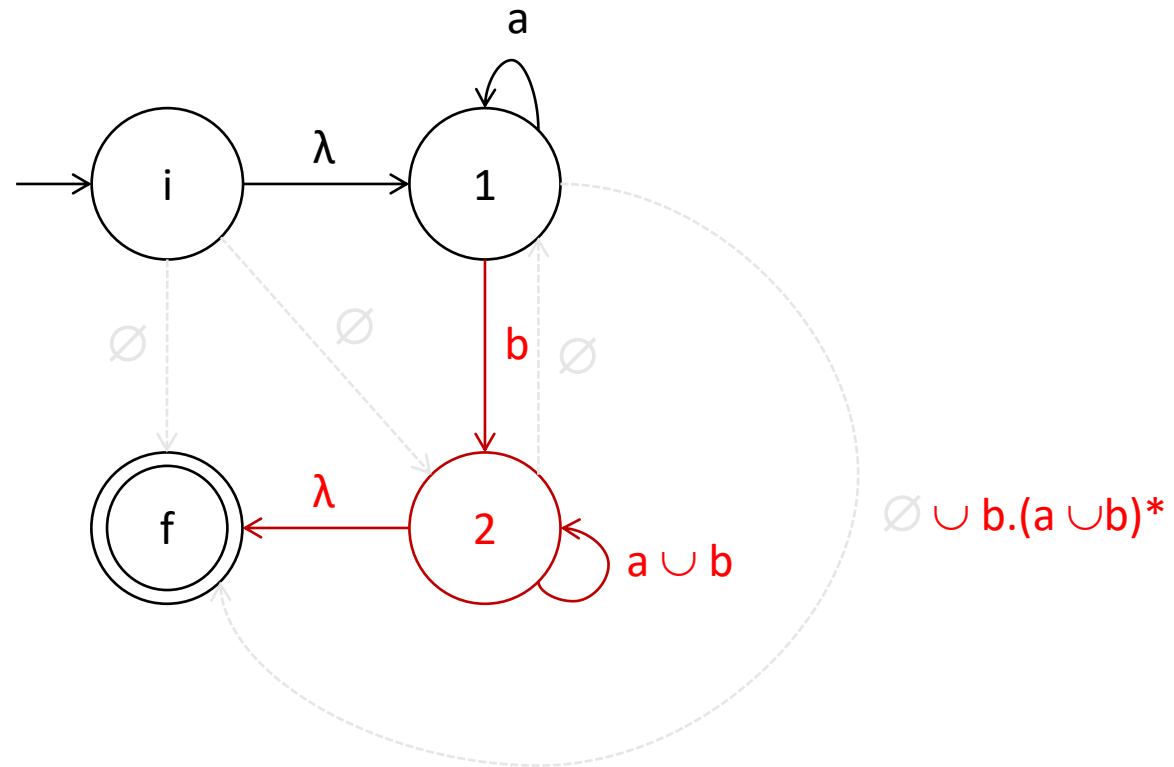
De AFNGs para expressões regulares



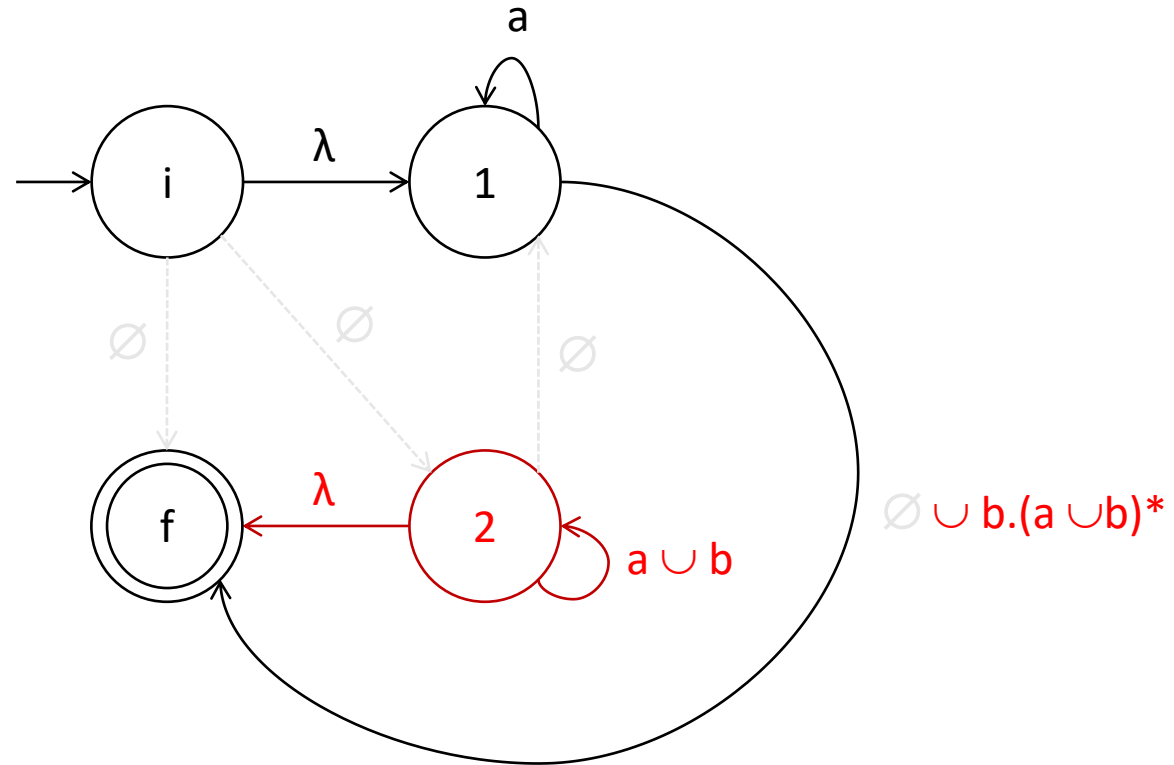
De AFNGs para expressões regulares



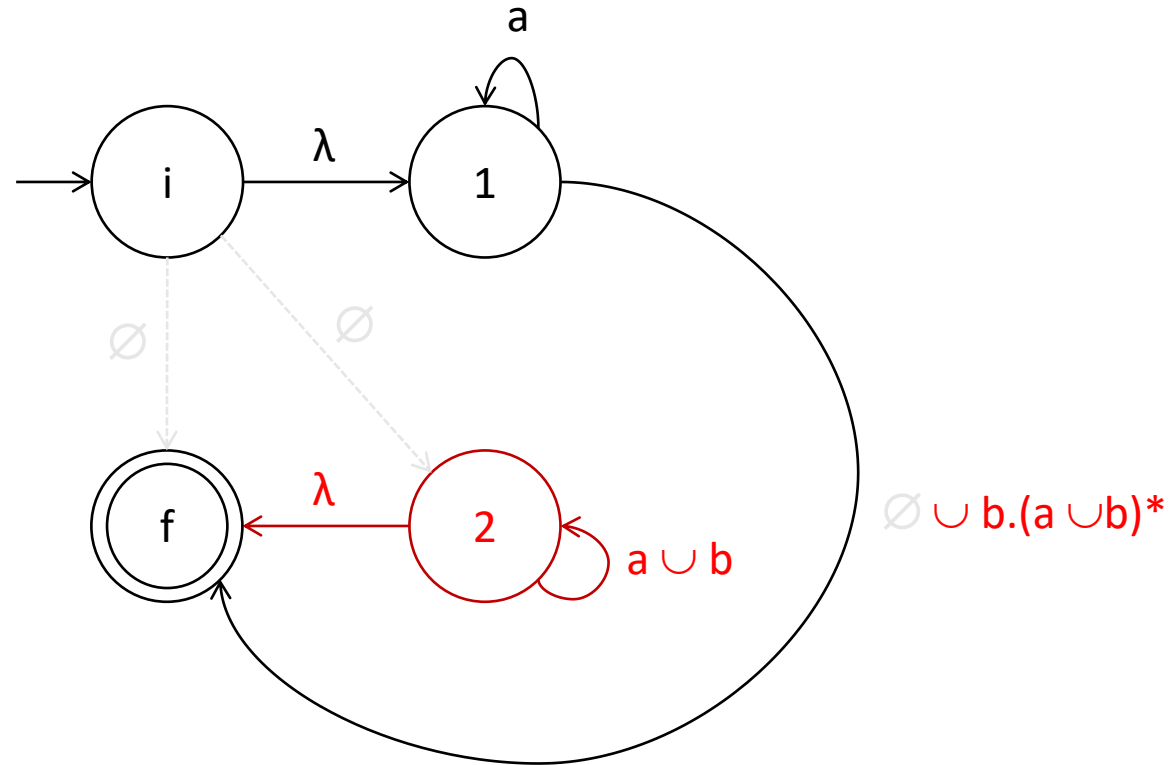
De AFNGs para expressões regulares



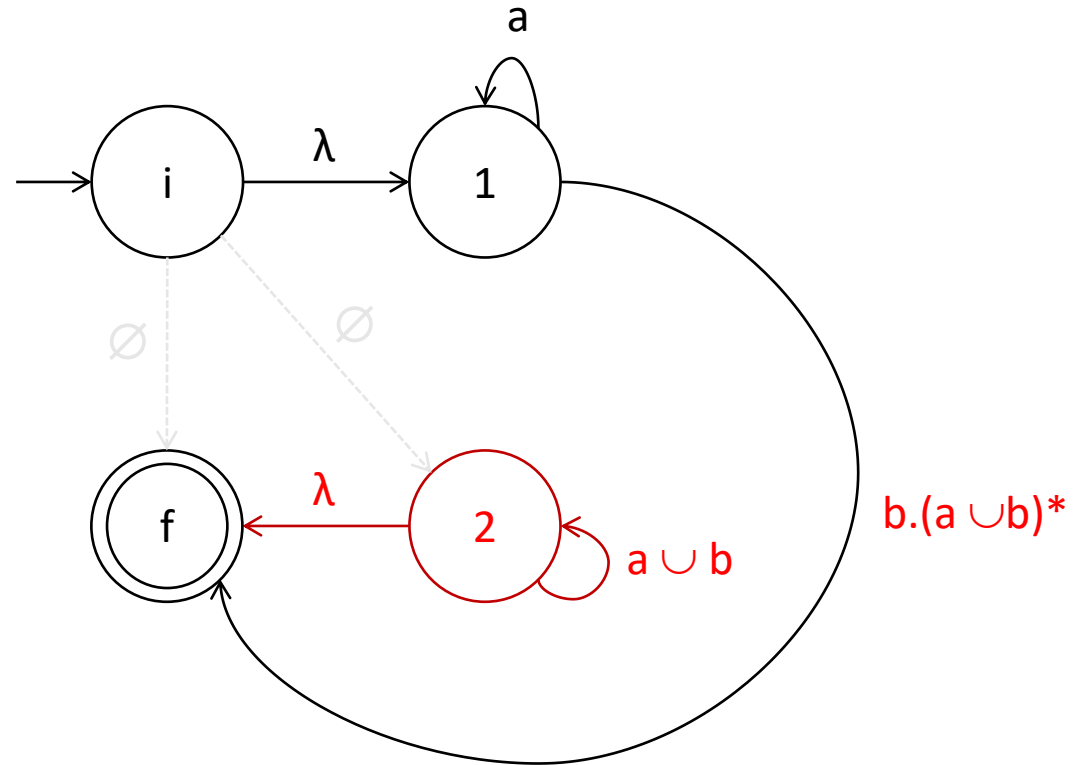
De AFNGs para expressões regulares



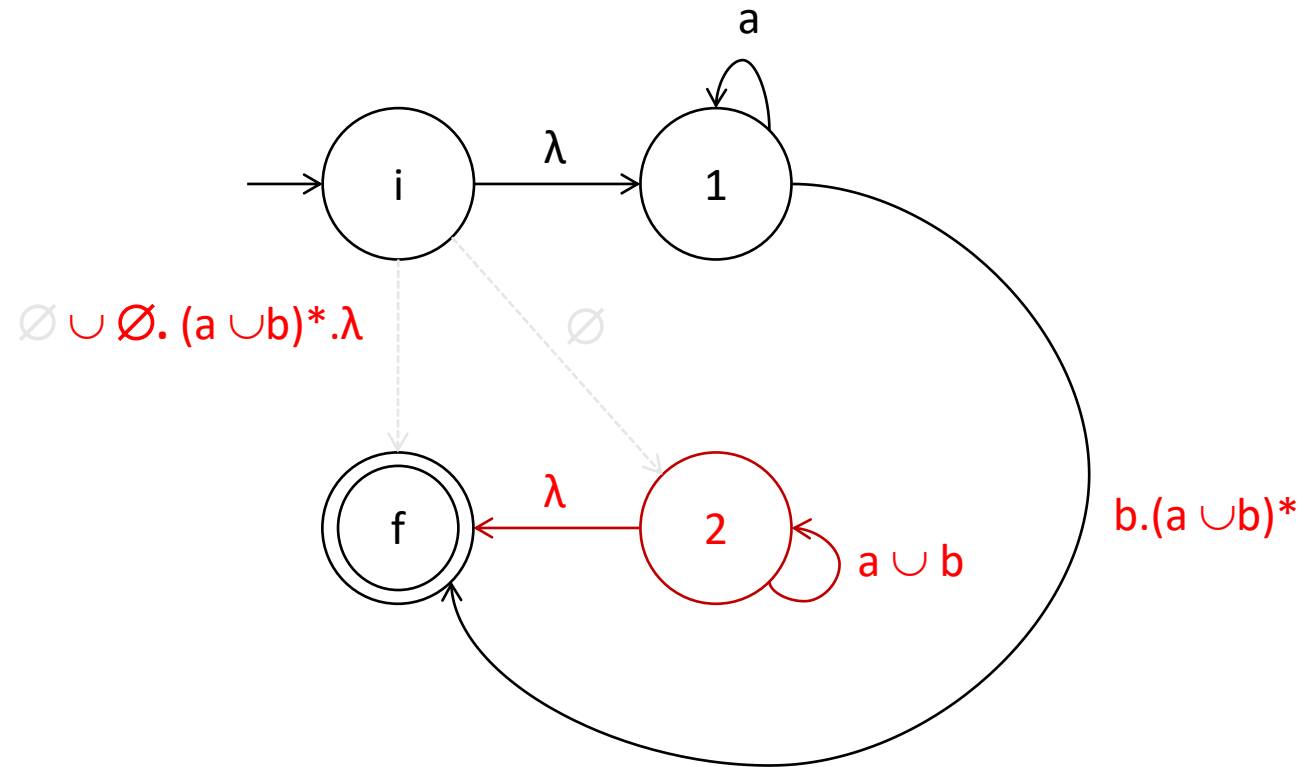
De AFNGs para expressões regulares



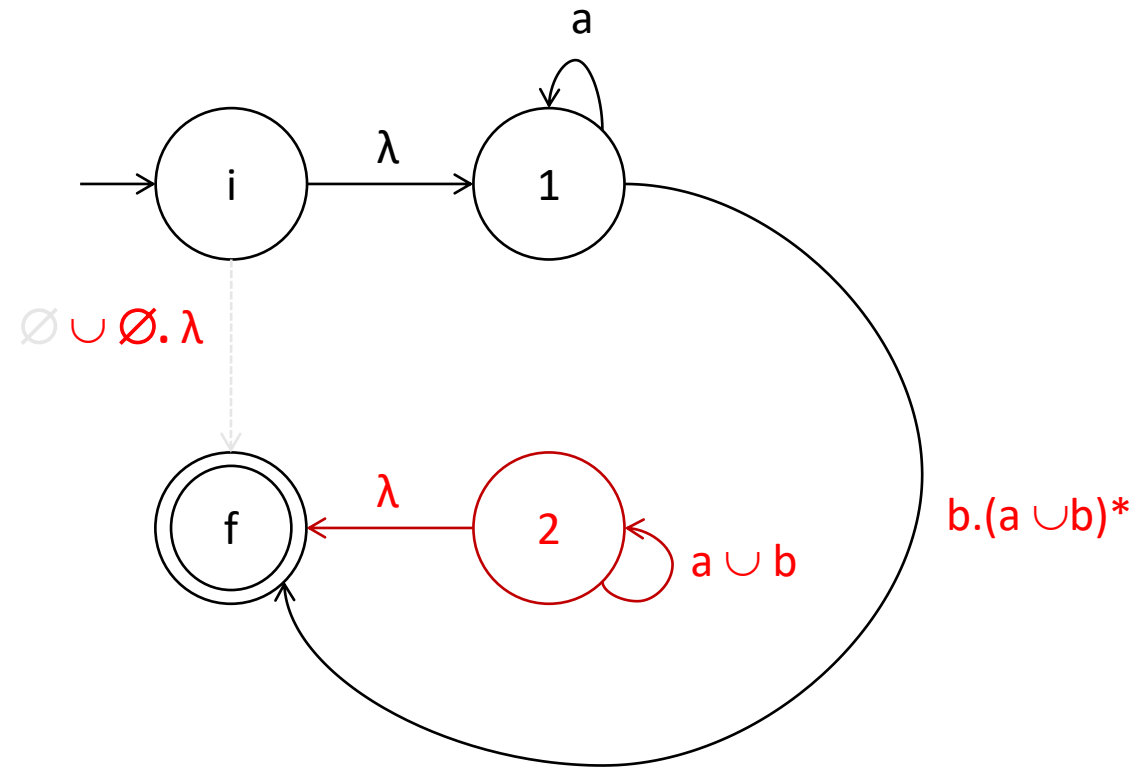
De AFNGs para expressões regulares



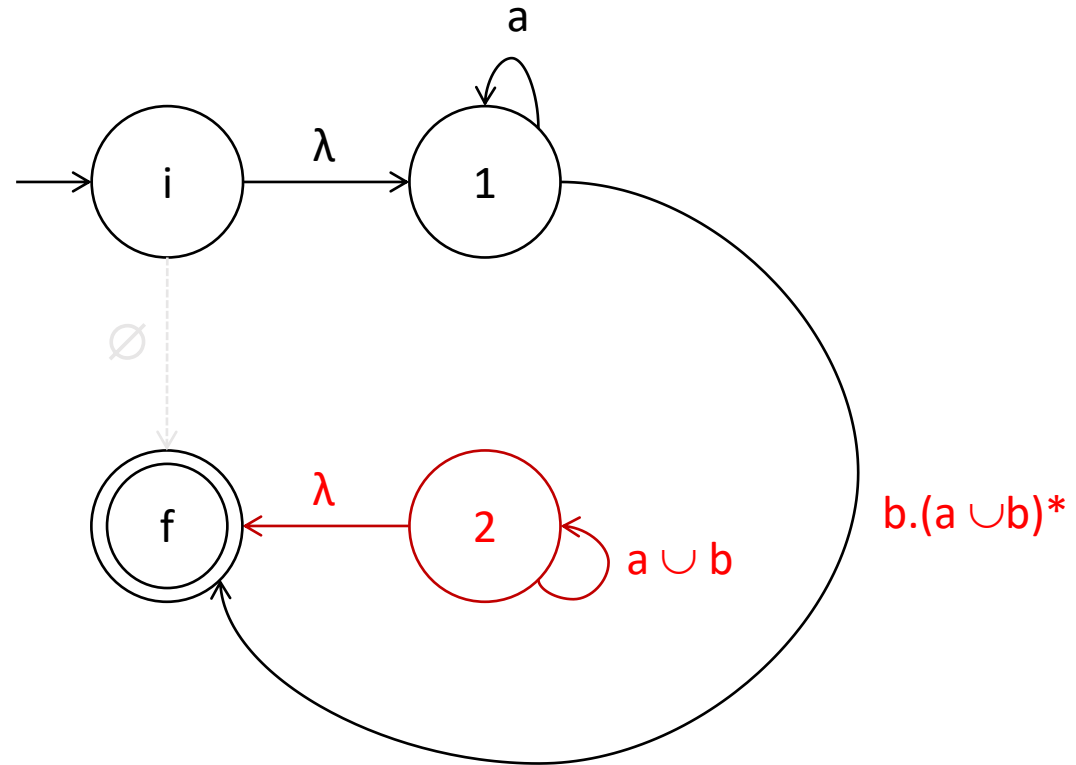
De AFNGs para expressões regulares



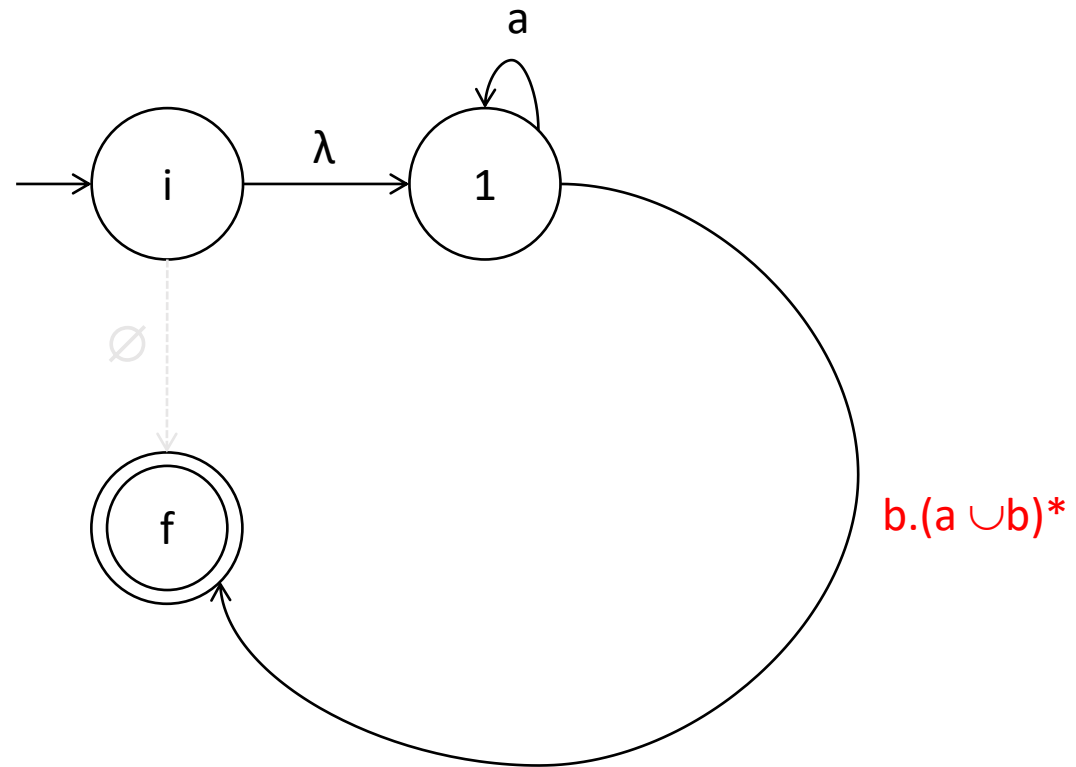
De AFNGs para expressões regulares



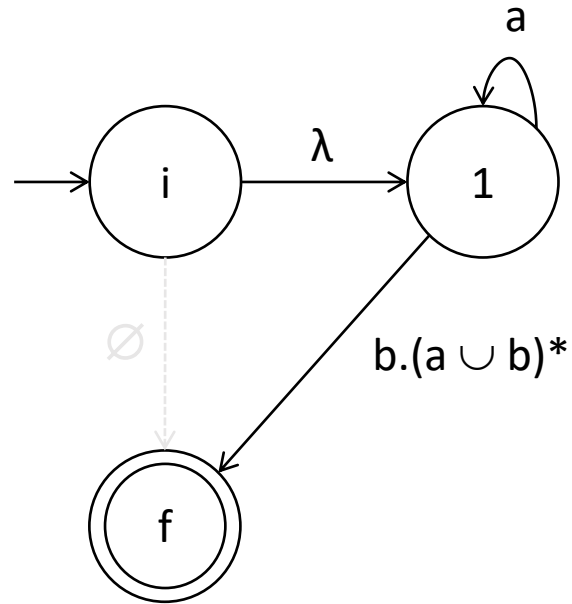
De AFNGs para expressões regulares



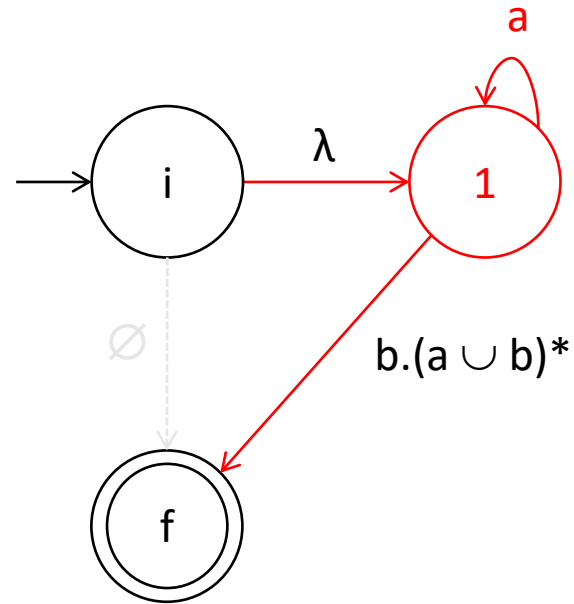
De AFNGs para expressões regulares



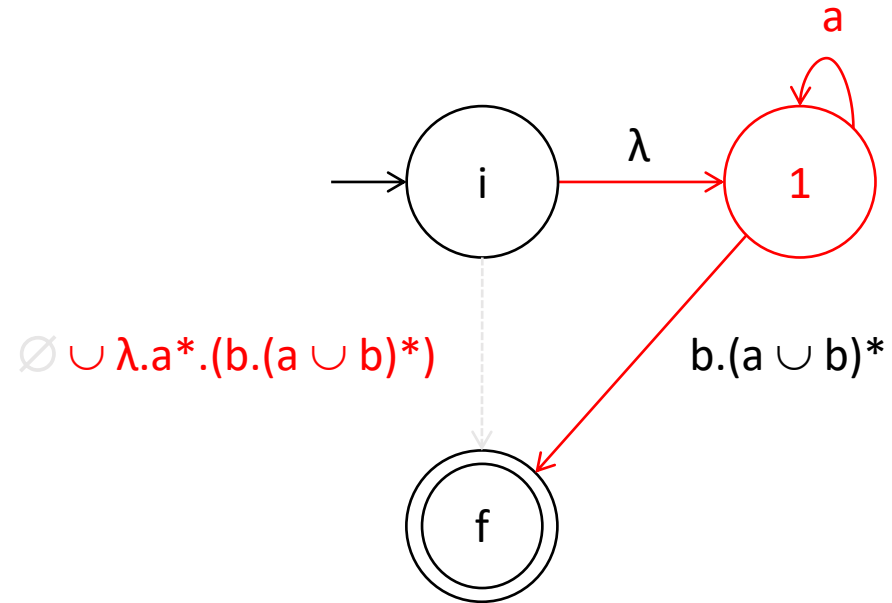
De AFNGs para expressões regulares



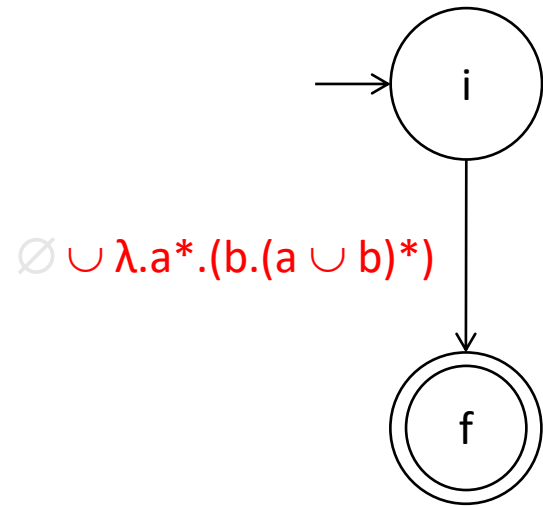
De AFNGs para expressões regulares



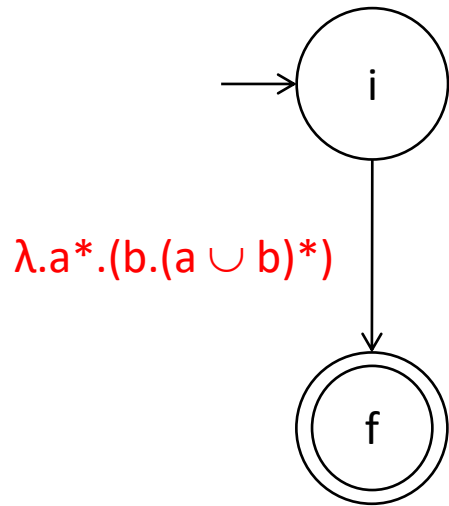
De AFNGs para expressões regulares



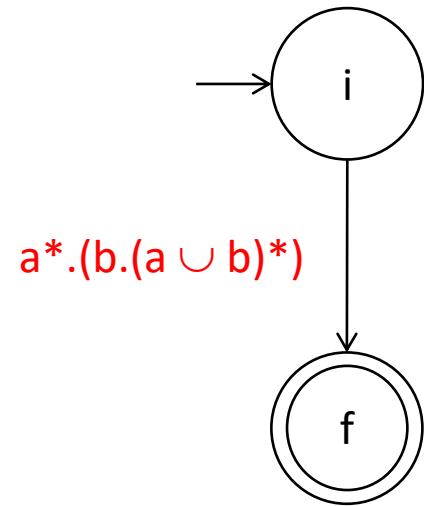
De AFNGs para expressões regulares



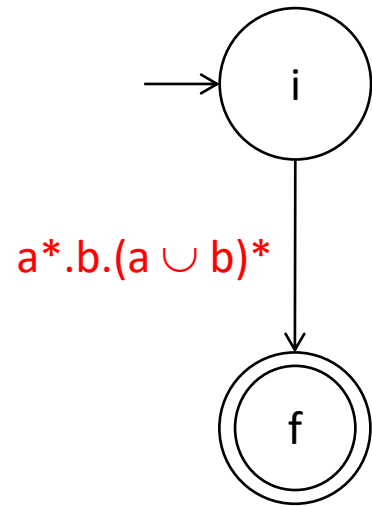
De AFNGs para expressões regulares



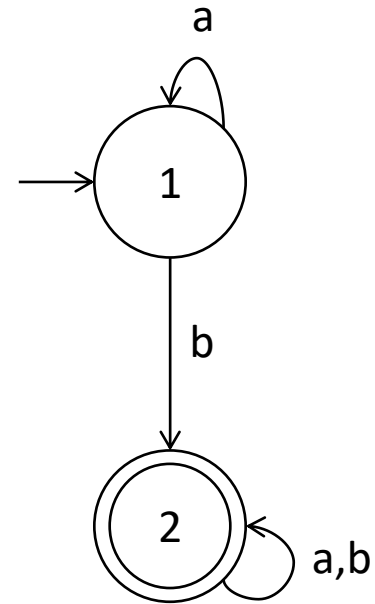
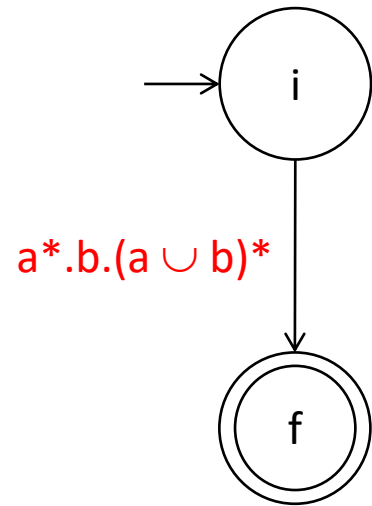
De AFNGs para expressões regulares



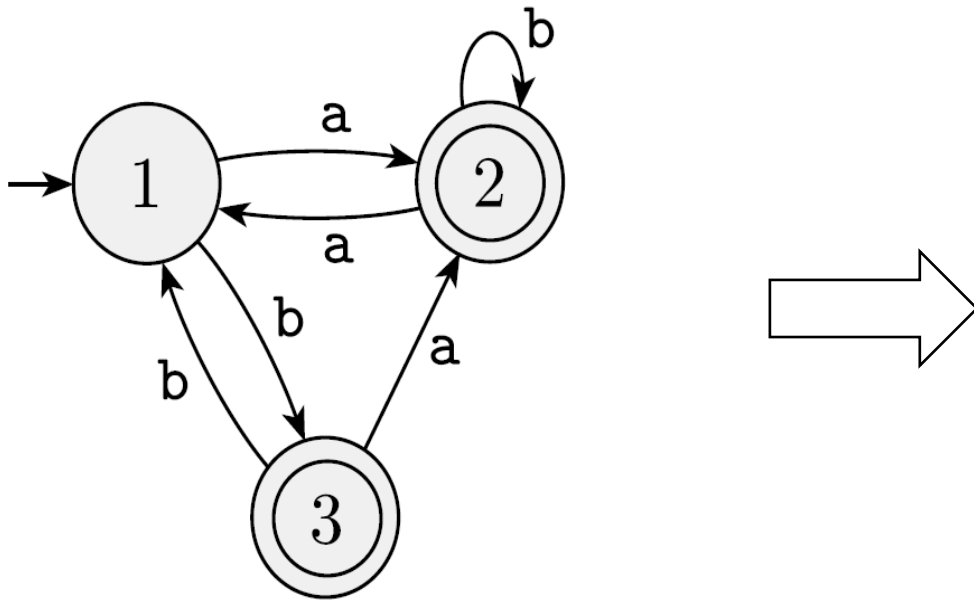
De AFNGs para expressões regulares



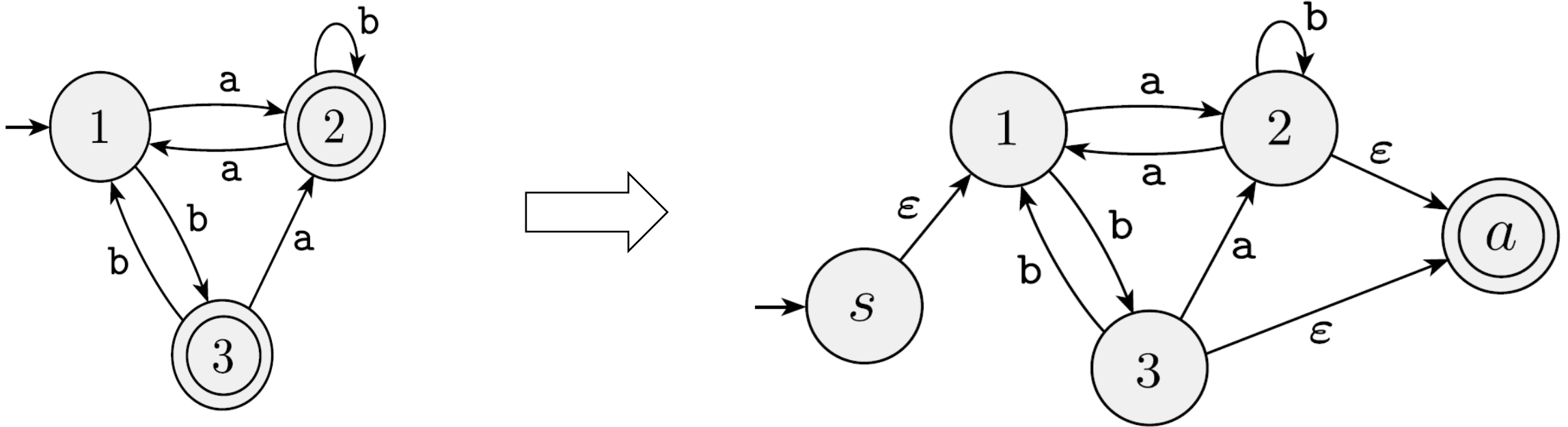
De AFNGs para expressões regulares



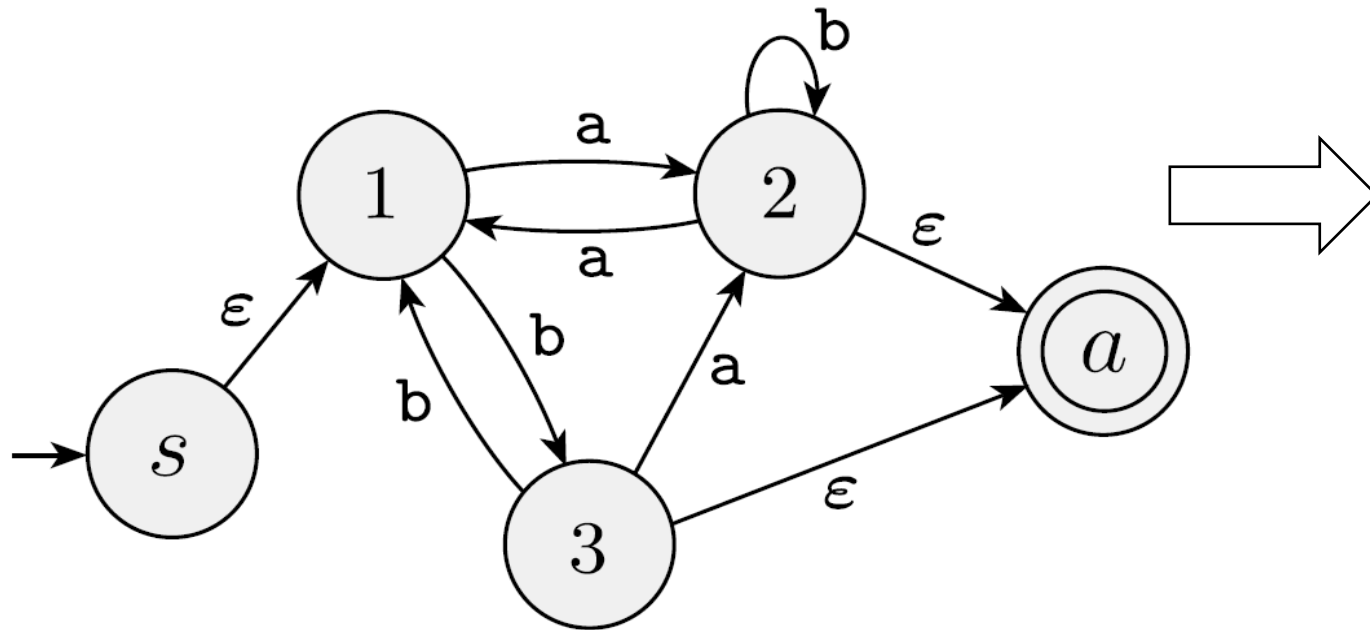
Outro exemplo: Geração do AFND



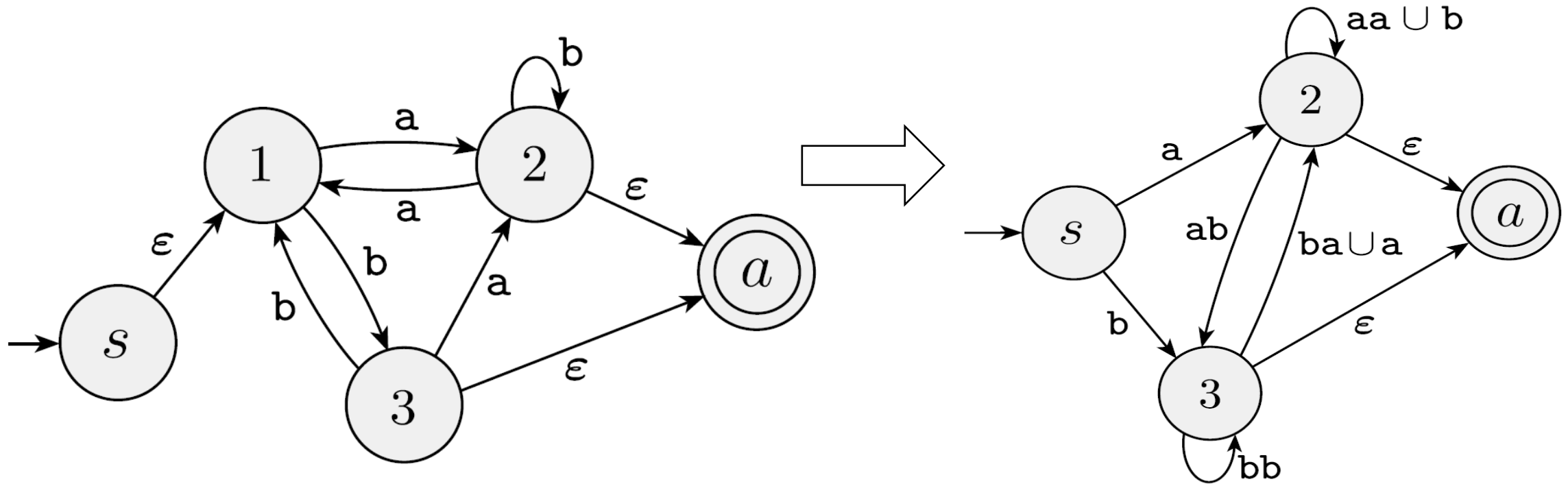
Outro exemplo: Geração do AFND



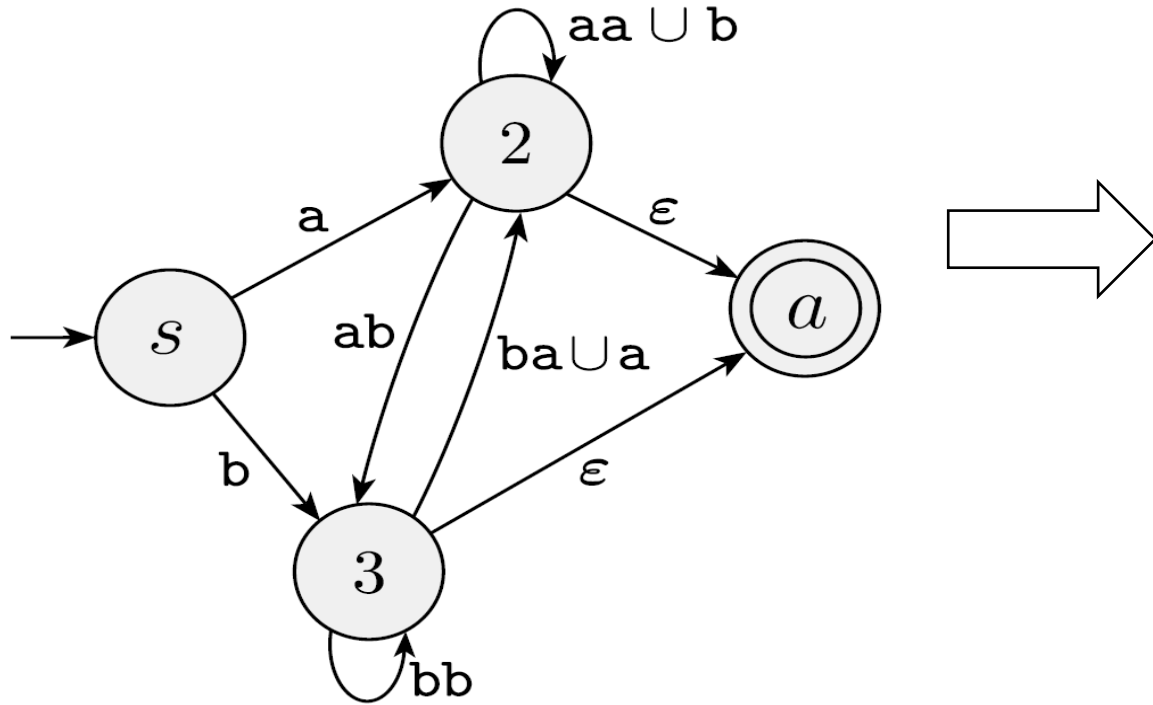
Removendo o estado 1



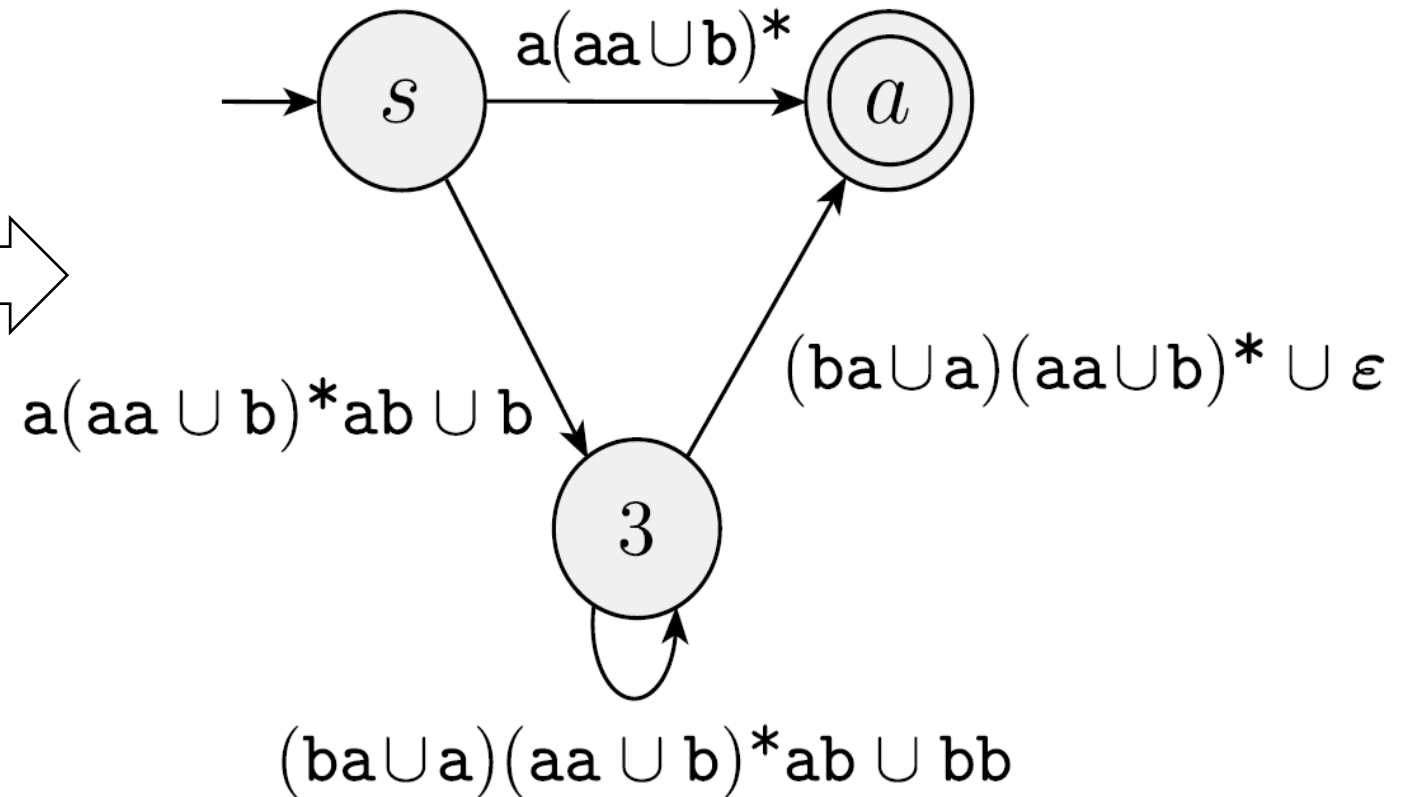
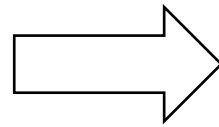
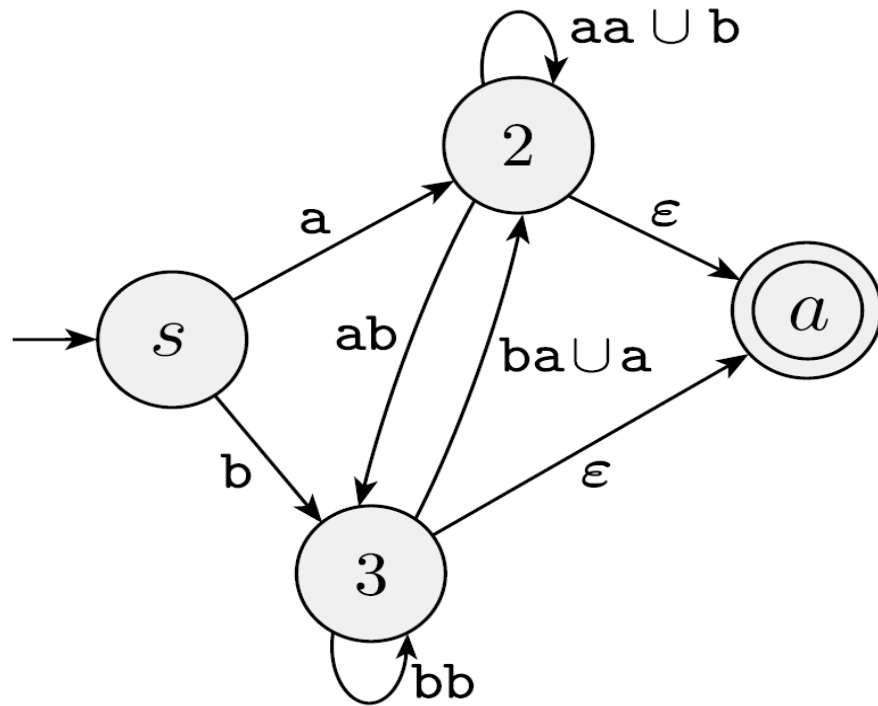
Removendo o estado 1



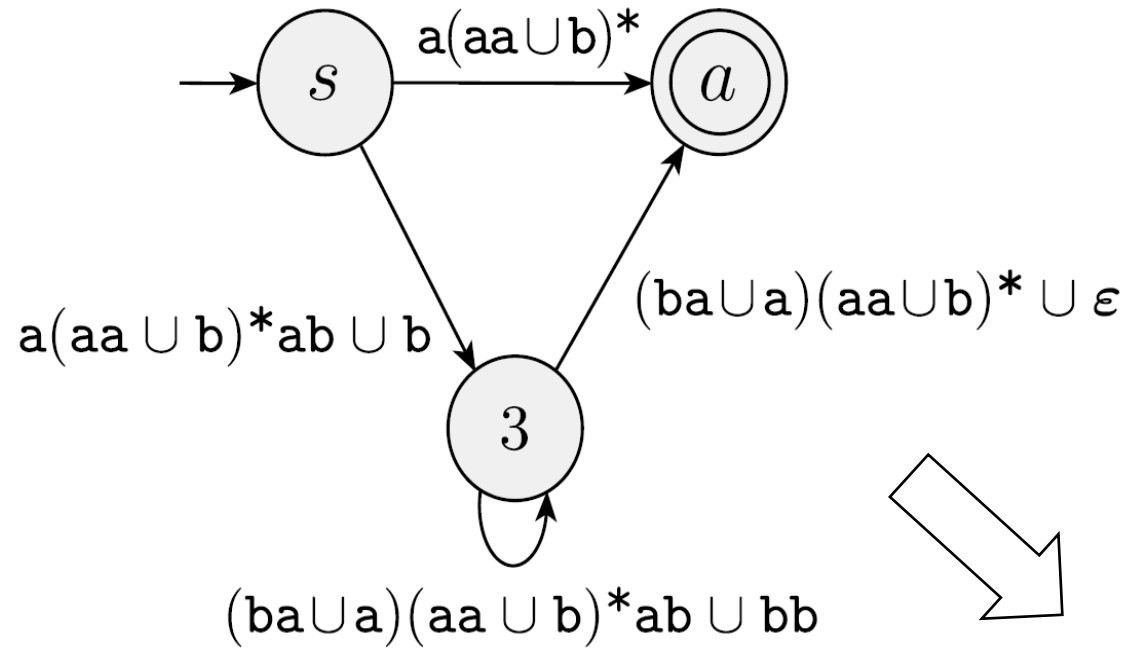
Removendo o estado 2



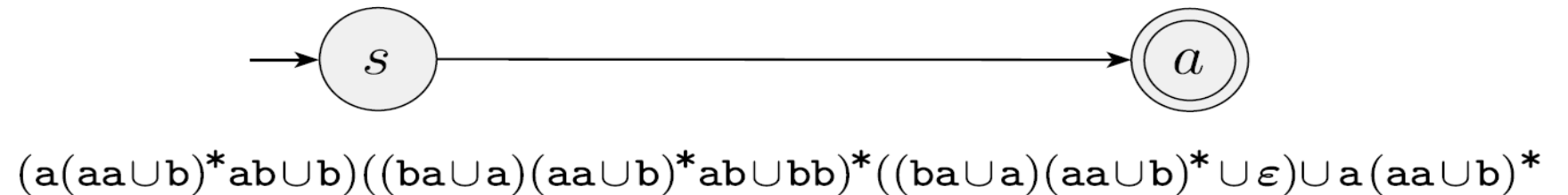
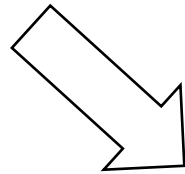
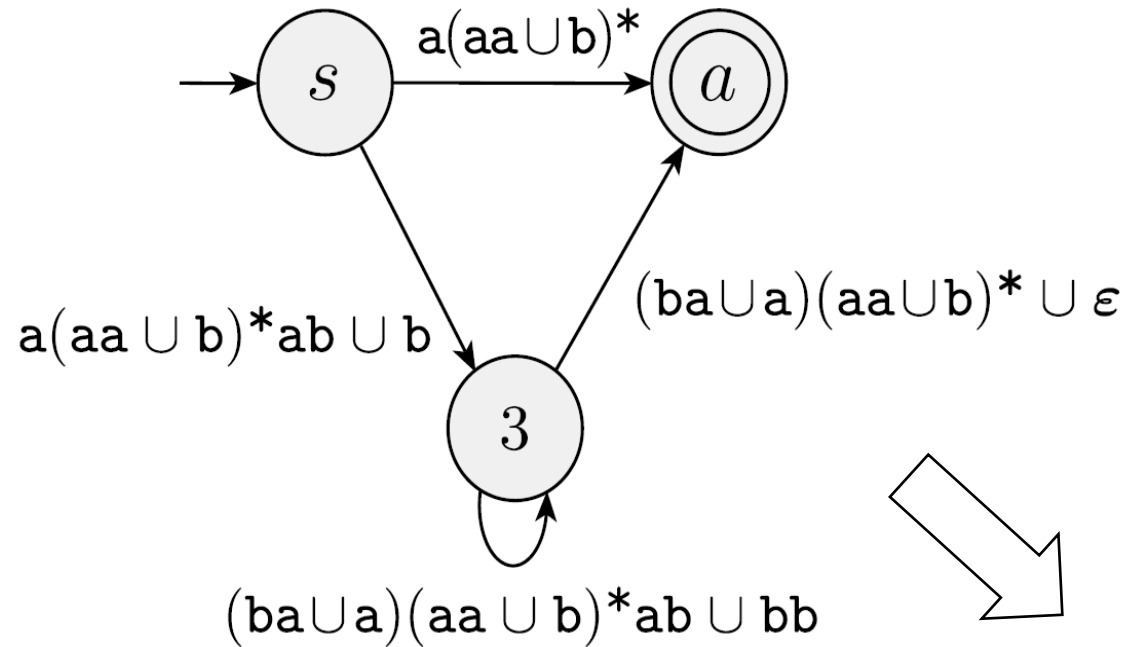
Removendo o estado 2



Removendo o estado 3



Removendo o estado 3



Linguagens não-regulares e lema
do bombeamento

Linguagens não-regulares

- Existem linguagens que não são regulares? O que isso significa?

Linguagens não-regulares

- Existem linguagens que não são regulares? O que isso significa?
- Qual a essência da não regularidade? Tamanho da linguagem?

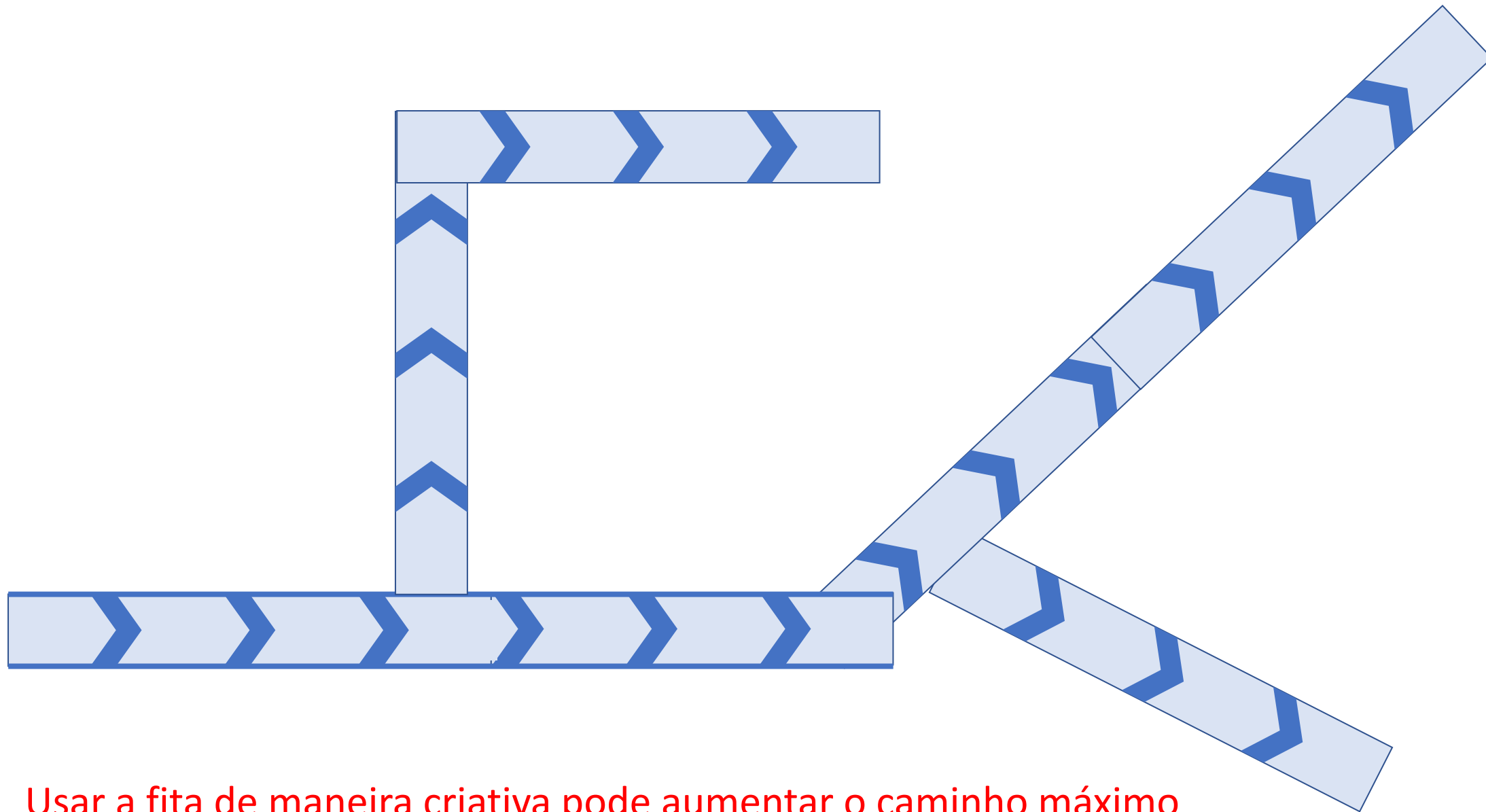
Linguagens não-regulares

- Existem linguagens que não são regulares? O que isso significa?
- Qual a essência da não regularidade? Tamanho da linguagem?
- Considerando expressões que não seguem a definição de expressões regulares, por exemplo 0^n1^n , é possível fazer um autômato finito?
- Conclusão: Existe uma certa limitação no tipo de padrões que os autômatos finitos conseguem reconhecer...

Lema do bombeamento



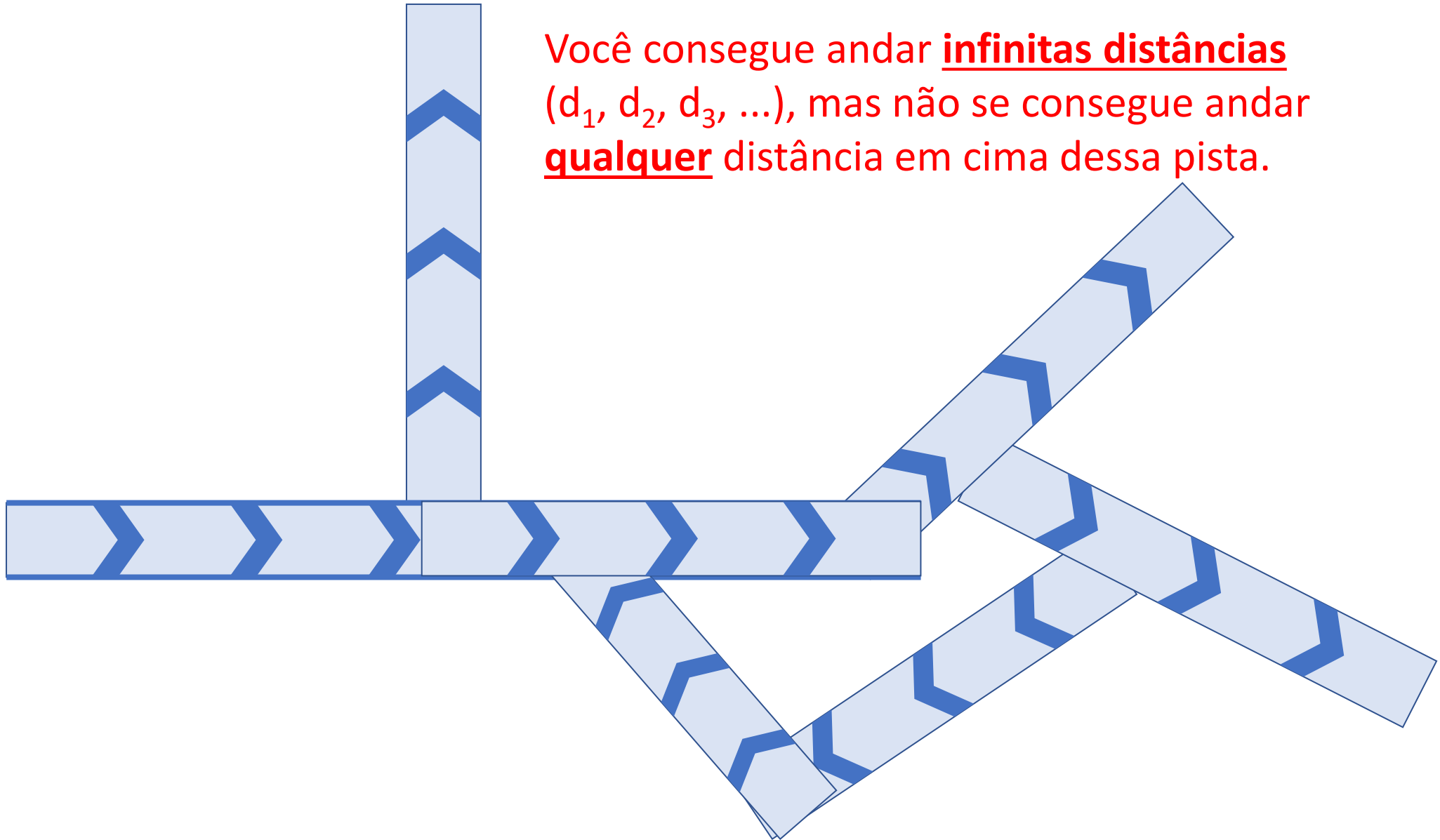
- O exemplo da fita
 - Use uma fita para criar um caminho
 - Você pode cortar pedaços da fita para criar ramificações
 - Marque um ponto de partida e de fim
 - Você só pode andar na direção marcada na fita
- Dada uma fita de comprimento 1 Km: Qual o maior caminho que você consegue percorrer em cima da fita? E se alguém afirma que percorreu uma distância maior?



Usar a fita de maneira criativa pode aumentar o caminho máximo que alguém conseguiria percorrer em cima dela?

Usar a fita de maneira criativa pode aumentar o caminho máximo que alguém conseguiria percorrer em cima dela? Até que sim...

Você consegue andar infinitas distâncias (d_1, d_2, d_3, \dots), mas não se consegue andar qualquer distância em cima dessa pista.

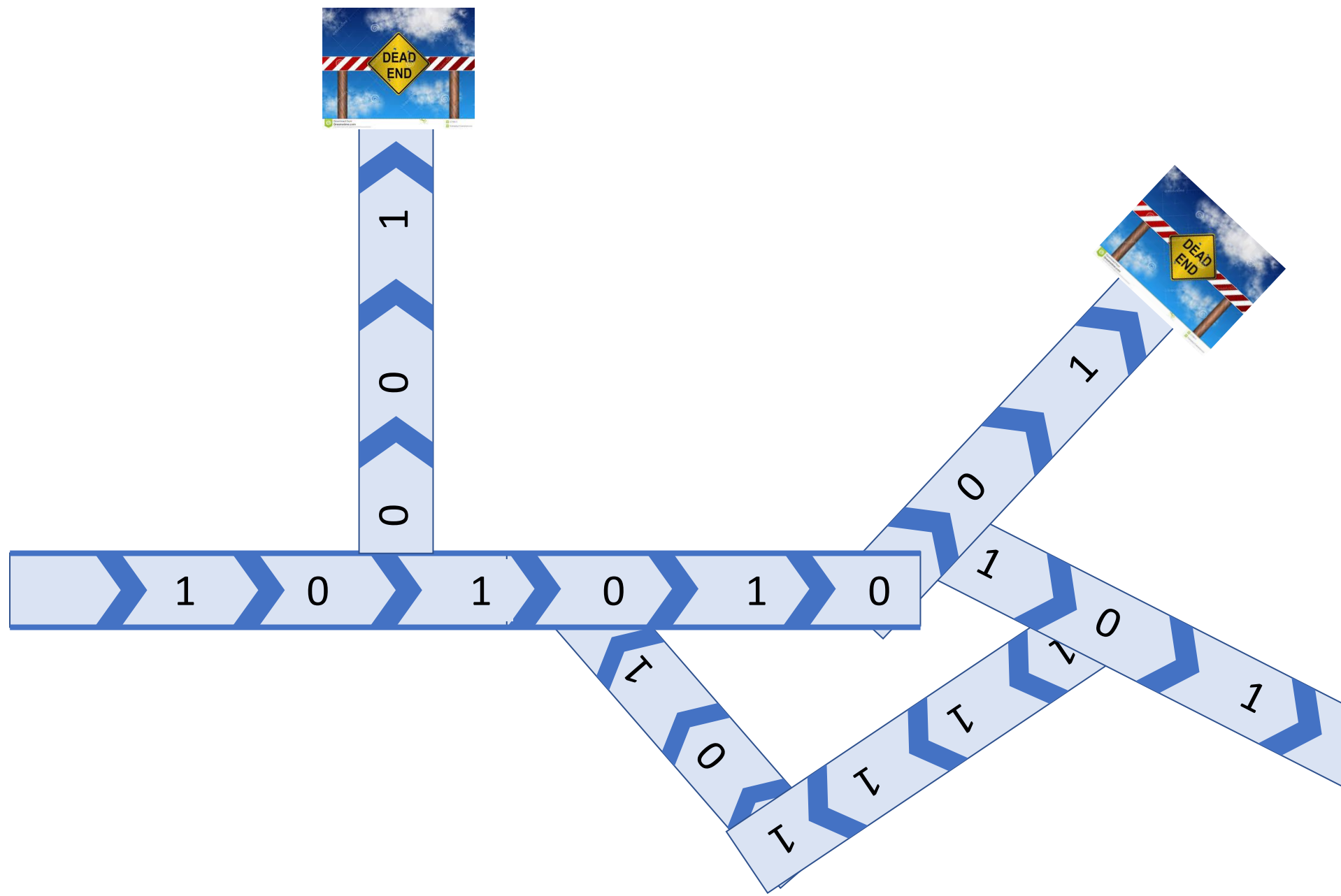


Lema do bombeamento (2)

- Agora a fita tem um símbolo associado a cada pequeno pedaço...

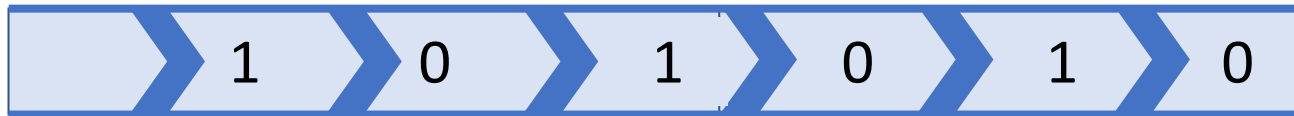


- Como antes você vai construir um caminho usando essa fita
 - Ou melhor, um labirinto



Lema do bombeamento (2)

- Agora a fita tem um símbolo associado a cada pequeno pedaço...

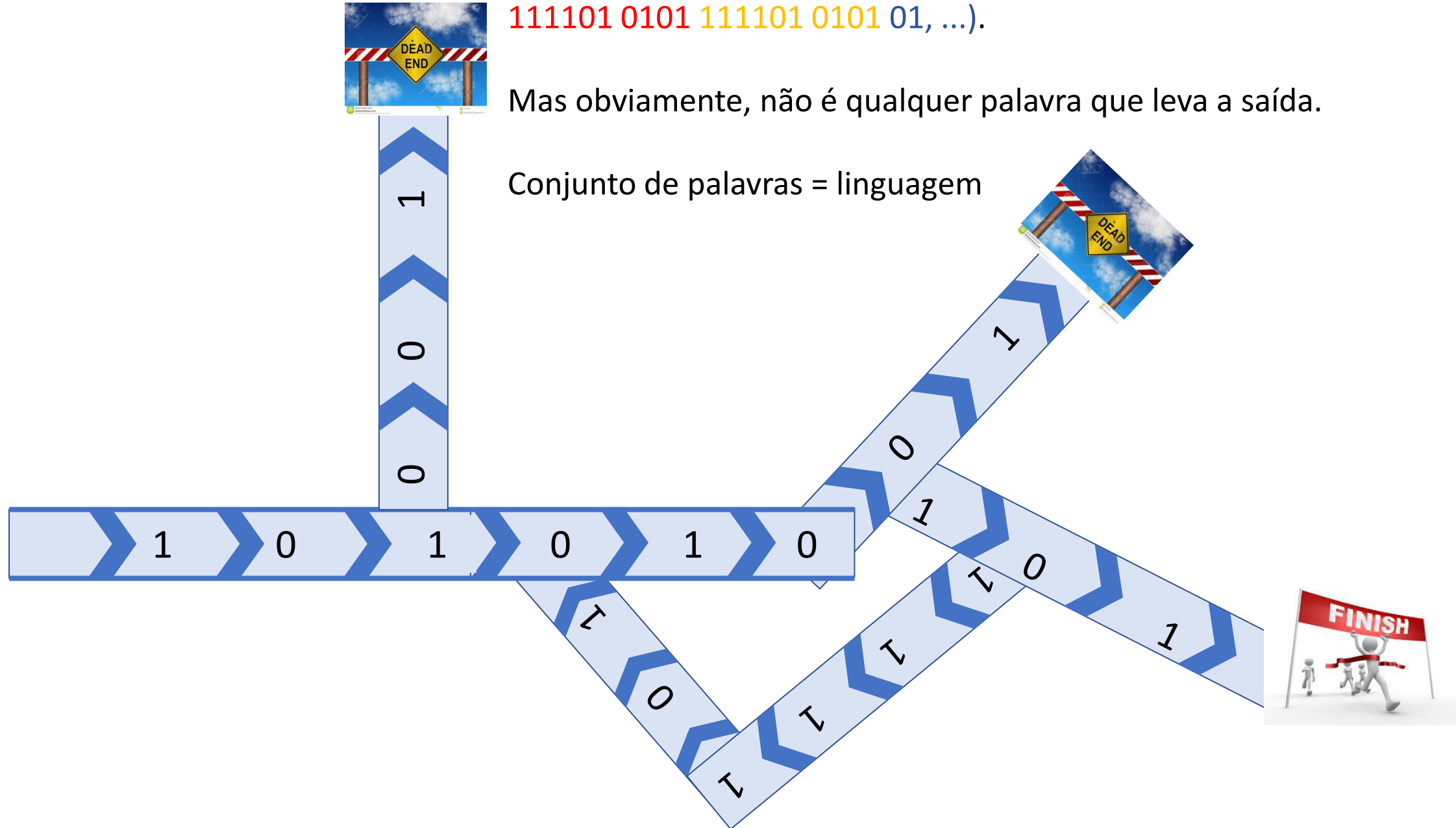


- Como antes você vai construir um caminho usando essa fita
 - Ou melhor, um labirinto...
 - Uma sequência de 0s e 1s pode ser uma “solução” para identificar um caminho você leve à saída
 - Obviamente, nem todas as sequências levam à saída (nem todas as sequências são soluções)

Neste labirinto, não existe um único mapa levando à saída, existem infinitos (ex., 101010101, 101010 1 111101 0101 01, 101010 1 111101 0101 111101 0101 0101 01, ...).

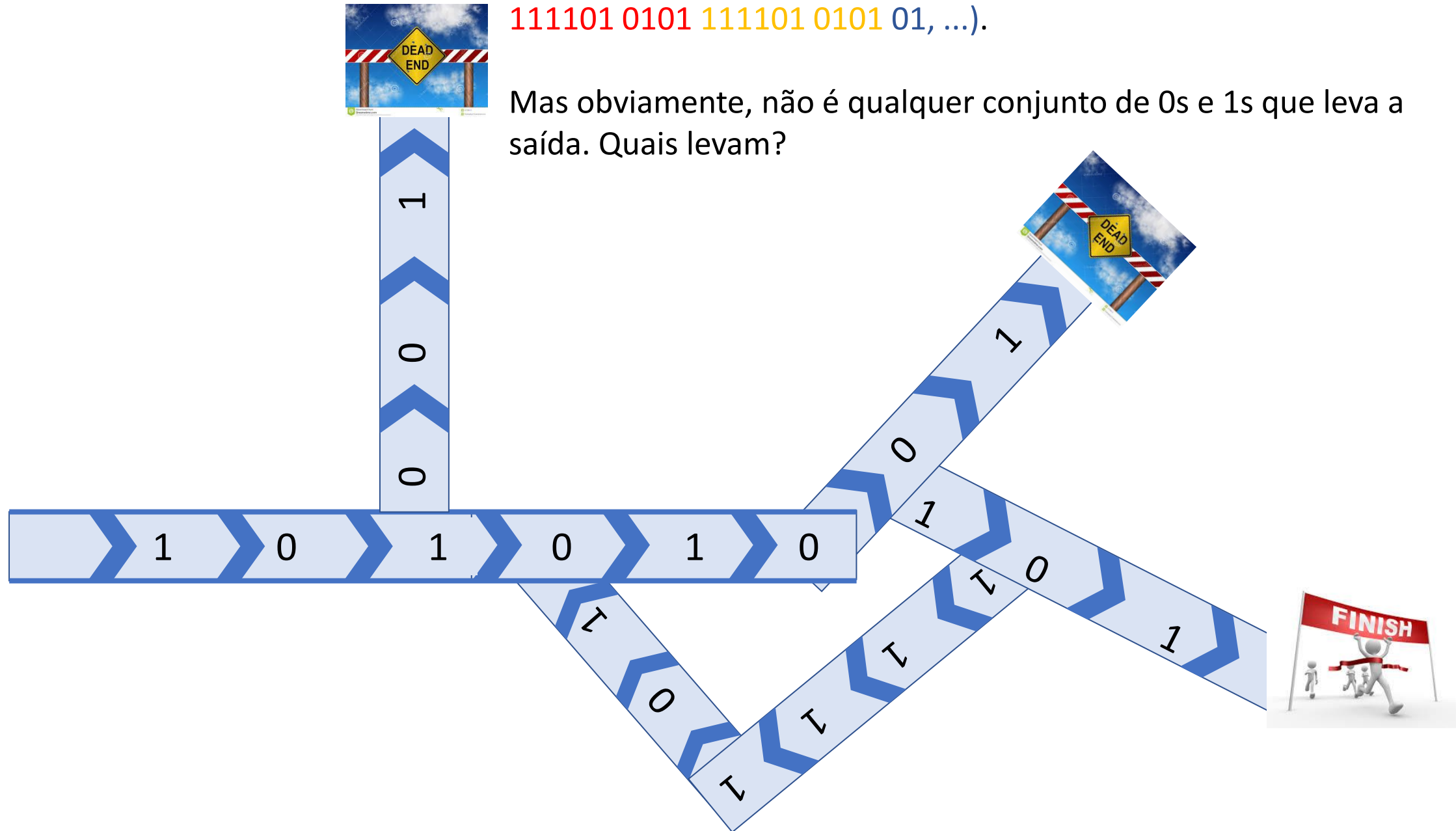
Mas obviamente, não é qualquer palavra que leva a saída.


Conjunto de palavras = linguagem



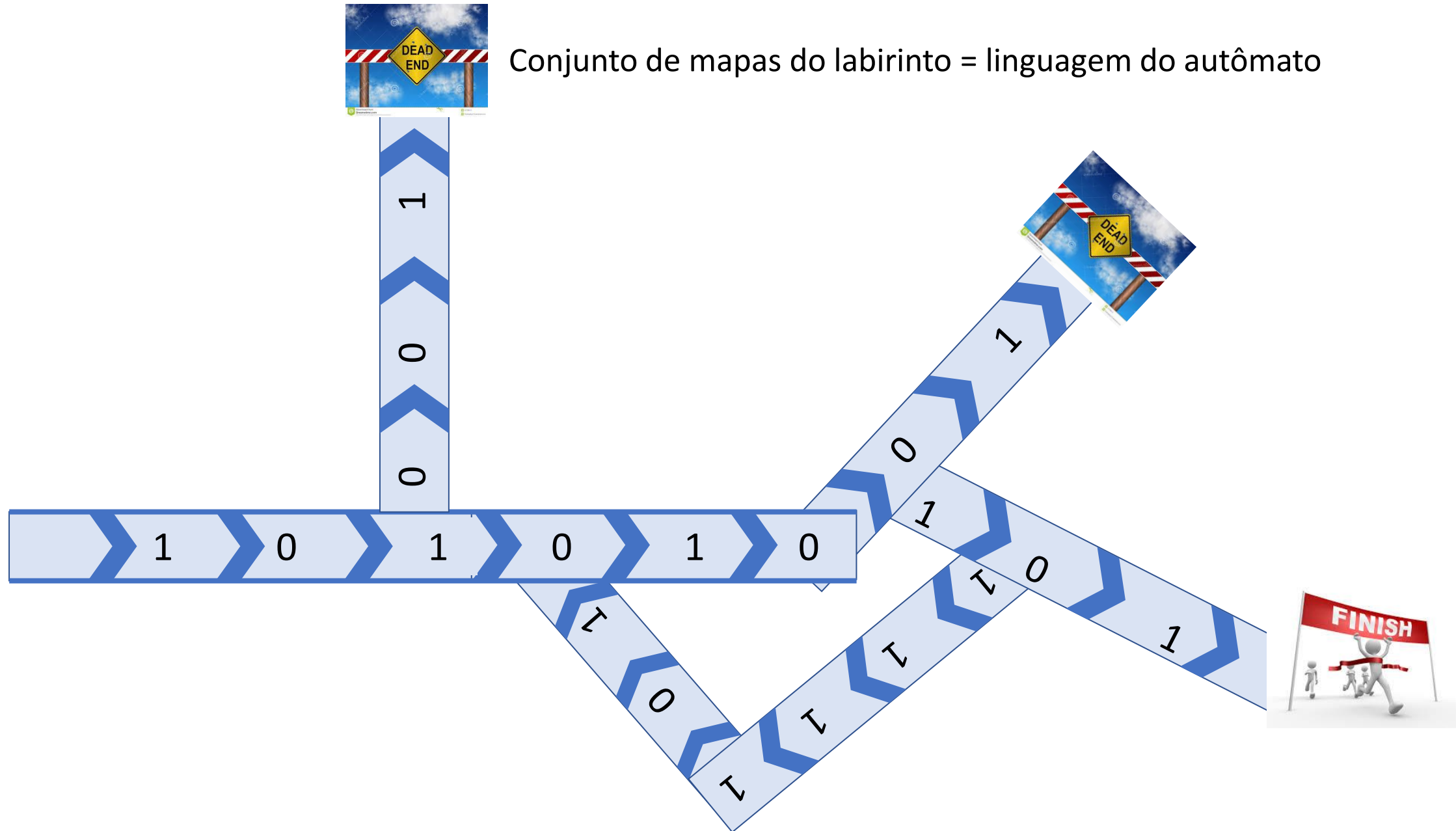
Neste labirinto, não existe um único mapa levando à saída, existem infinitos (ex., 101010101, 101010 1 111101 0101 01, 101010 1 111101 0101 111101 0101 0101 01, ...).

Mas obviamente, não é qualquer conjunto de 0s e 1s que leva a saída. Quais levam?



Olhando para o labirinto como um autômato: cada  é um estado; um mapa é uma palavra aceita.

Conjunto de mapas do labirinto = linguagem do autômato



Lema do bombeamento

- Intuição
 - Para uma linguagem ser infinita, ela precisa ter palavras de comprimento arbitrário, já que o alfabeto é finito
 - Se ela é regular, então algum padrão precisa se repetir
 - Caso contrário, a “fita” (o autômato) precisaria ser infinita
- Existe um comprimento de palavra p (o mapa que diz como sair)
 - Seja p um tamanho arbitrário para a fita usada no meu labirinto
 - Talvez palavras menores que p não precisem passar pelo ciclo
 - Mas certamente as maiores que p terão que ter passado por um ciclo

Toda palavra precisa passar pelo ciclo?

Quais palavras têm que passar obrigatoriamente pelo ciclo?

