Autômatos finitos

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

Na aula passada – Máquinas de estado

- **Alfabeto**: símbolos que podem ser usados como entrada (de nossa máquina abstrata)
  - Ex.: {B}; {1, 2}; {a, b, c, d, ..., z}; {Botão1_pressionado, Botão1_liberado}
- **Cadeia** (palavra, string): concatenação finita de símbolos do alfabeto em uso
  - Ex.: 11; 121212; PPPPPP
- **Diagrama de estados**: o "programa" da nossa máquina, com estados e **transições** ligando esses estados
- **Estado**: posição no diagrama; porção relevante da história de execução da máquina
- **Estado inicial**: estado onde a máquina começa

Máquina de estados como um autômato

- Como discutido na aula anterior, nosso objetivo não é só rastrear uma execução, mas sim implementar soluções para problemas de decisão

- Um autômato finito pode ser descrito como uma máquina de estados que diz sim ou não
  - Processa a cadeia de entrada, gerando uma resposta no final
  - Estado final: marca onde a máquina precisa terminar para que a resposta seja sim

- Um exemplo: a máquina que verifica a pontuação (súmula)

Verificação de pontuação como um autômato finito

- Ao contrário do interruptor, nem toda sequência de entrada faz sentido (ou seja, representa um jogo válido)
  - Exemplo 1: 7-0 não é um placar válido no nosso jogo, logo a cadeia de entrada AAAAAAA não é uma cadeia válida
  - Exemplo 2: 2-4 é um placar válido no nosso jogo, logo a cadeia de entrada ABABBB é uma cadeia válida

- A contadora de pontos como um autômato testa uma propriedade da entrada
  - Neste caso, testa se é um placar válido
  - Aceita somente cadeias que fazem sentido para aquele esporte

Outros conceitos – Autômatos finitos

- **<u>Estado final</u>**: marca onde a máquina precisa terminar para que a resposta seja sim (zero, um ou vários estados finais)

- **<u>Máquina de estados</u>** vs. **<u>autômato finito</u>**
  - Um diagrama de estados serve para acompanhar o estado de um sistema
  - AF é uma máquina que "resolve" algum problema aceitando ou rejeitando uma "cadeia" que foi passada como entrada

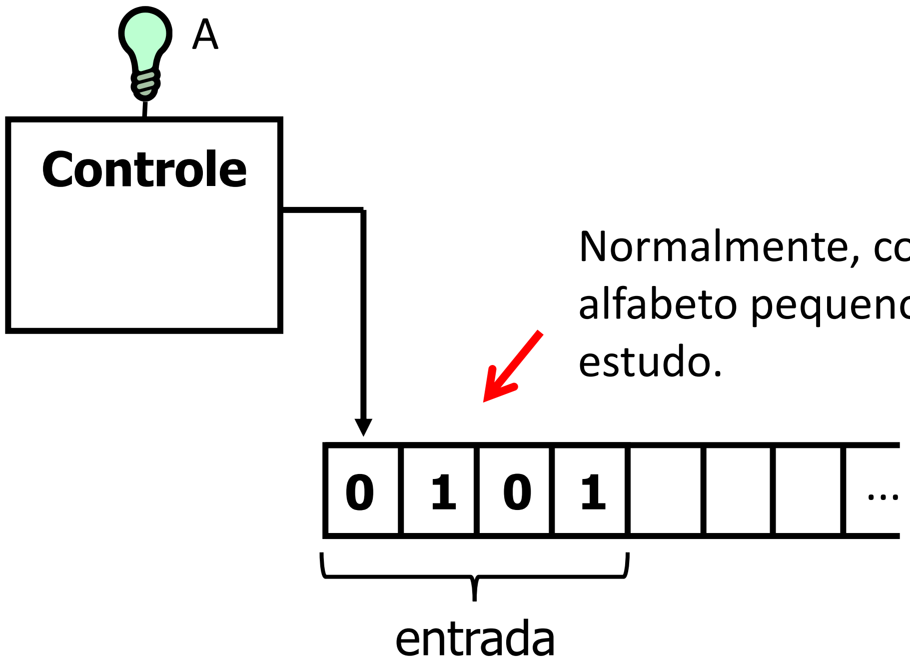
- **<u>Linguagem</u>**: um conjunto de cadeias que fazem uma máquina retornar SIM
  - Exemplo para o reconhecedor de súmulas: {AAA, ABAA, BBB, ABBB, ...}
  - Cada autômato **<u>reconhece</u>** exatamente uma linguagem

- Essa forma de "resolver" problemas, aceitando ou rejeitando, mostra uma das simplificações que faremos:
  - Trataremos de problemas de decisão (sim/não, verdadeiro/falso)

Autômatos finitos

Autômato finito

- Assumimos que os AFs são compostos de uma unidade de controle e uma fita, onde eles leem as entradas



Normalmente, consideramos um alfabeto pequeno. Isso facilita o nosso estudo.

Autômato Finito (2)

- Resolve um problema processando uma entrada e respondendo SIM ou NÃO
  - Processar: tratar os símbolos da cadeia de entrada um a um
  - SIM: Aceitação, ao terminar de processar os símbolos, o autômato está em um dos estados finais
  - NÃO: Rejeição, ao terminar de processar os símbolos, o autômato está em um estado que não é final

- Nossa verificadora de pontuação
  - Aceitava súmulas válidas (ex., AAA, AABA, ABABABAA)
  - Rejeitava súmulas inválidas (ex., AA, BBBBBA, ABABAB)

Autômato Finito (3)

- Um autômato finito é definido através ...
  - Dos estados em que ele pode estar durante uma computação
  - Dos símbolos que podem ser usados para escrever as palavras de entrada (conjunto dos símbolos → alfabeto)
  - Das regras que descrevem como ele muda de estado para cada símbolo (transições)
  - Da indicação do estado inicial
  - Da lista dos estados que são considerados finais

Formalmente...

- Um AF M é uma 5-tupla <$Q, \sum, \delta, q_0, F$>
  - Q é um conjunto finito e não-vazio chamado de conjunto de estados
  - $\sum$ é um conjunto finito e não-vazio chamado de alfabeto
  - $\delta : Q \times \sum \rightarrow Q$ é a função (total) de transição do autômato
  - $q_0 \in Q$ é o estado inicial
  - $F \subseteq Q$ é o conjunto de estados finais

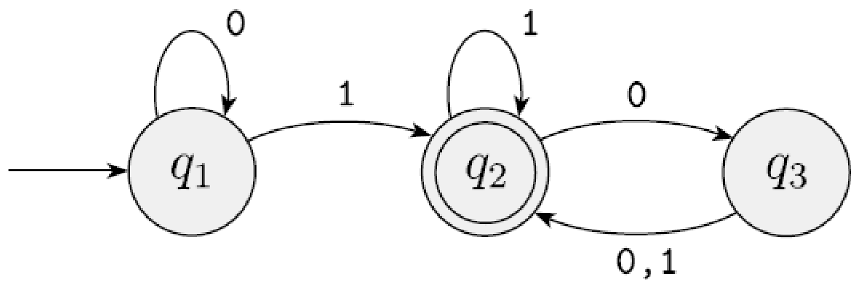
- Embora possa ser mais intimidadora, as definições formais são importantes por serem precisas e completas
  - (E é nosso objetivo aprender a lidar com o formalismo)

Exemplo

- Definição formal
  - 3 estados: $\{q_1, q_2, q_3\}$
  - Estado inicial: q_1
  - Estados finais: $\{q_2\}$
  - Alfabeto: {0,1}
  - Função de transição (regras):

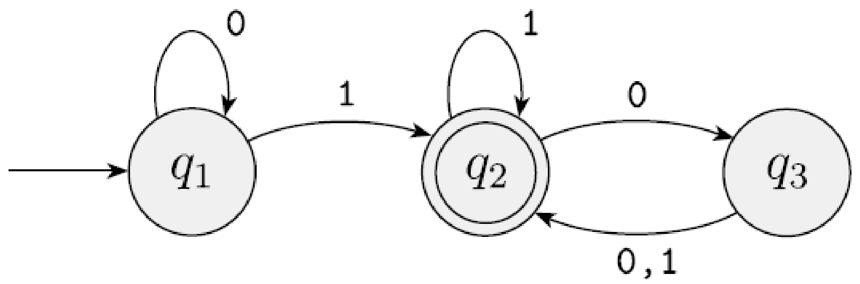
δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

- Diagrama de estados



Exemplo

- Que tipo de entrada faz este autômato sair do estado inicial e chegar no estado final?
- Diagrama de estados



Trabalhando com AFs - Recapitulando

- Que tipo de entrada faz o autômato sair do estado inicial e chegar no estado final?
  - Quais palavras fazem o autômato responder SIM?
  - Esta é a linguagem do AF
  - Linguagem: conjunto de palavras para as quais o autômato responde SIM → Aceita
  - Quando um autômato diz NÃO, ele rejeita a palavra

- De outra forma:
  - Cada autômato está associado a um conjunto (a linguagem)
  - Dada uma palavra qualquer, ele aceita se esta palavra estiver no seu conjunto

Trabalhando com AFs (2)

- O que queremos dos AFs?
  - Primeiro, saber como projetá-los para alguns problemas simples
  - Em geral, saber exatamente que tipo de problemas eles resolvem
  - Exemplo:
    - Existe um que consegue procurar uma “substring” dentro de uma entrada?
    - Existe um que consegue decidir se uma cadeia de entrada tem comprimento par?
    - Existe um que consegue distinguir se uma expressão algébrica é válida? Exemplo: a+b*((c+d)/a)

Desenvolvendo autômatos finitos

- Desenvolva um autômato que verifique a ocorrência do padrão 001 em uma entrada composta por símbolos 0 e 1
  - Olhando símbolo a símbolo, o autômato precisa sinalizar se a dada sequência aconteceu, usando apenas uma memória limitada (i.e., onde está no seu diagrama)
- Estratégia: se coloque no lugar do autômato e pense como ele reagiria

Desenvolvendo autômatos finitos

- Candidatos a estado: o que deve ser lembrado no final?
  - Vi a sequência (aceitação)
  - Não vi a sequência ainda

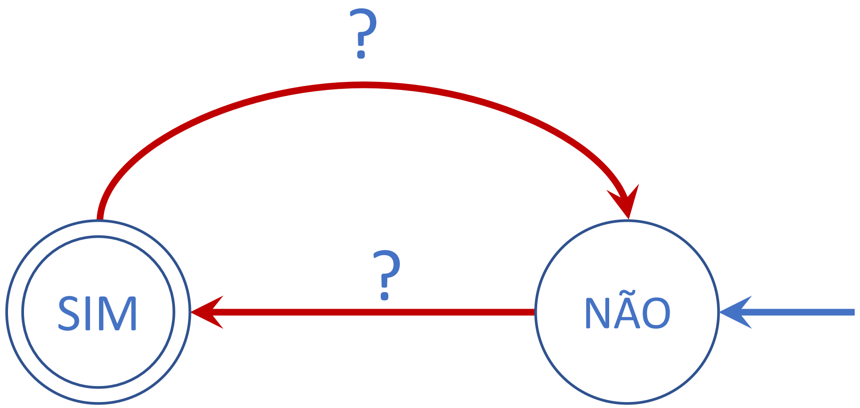
Nota: existem situações onde você vai precisar de vários estados finais.

Desenvolvendo autômatos finitos

- Como eu começo?

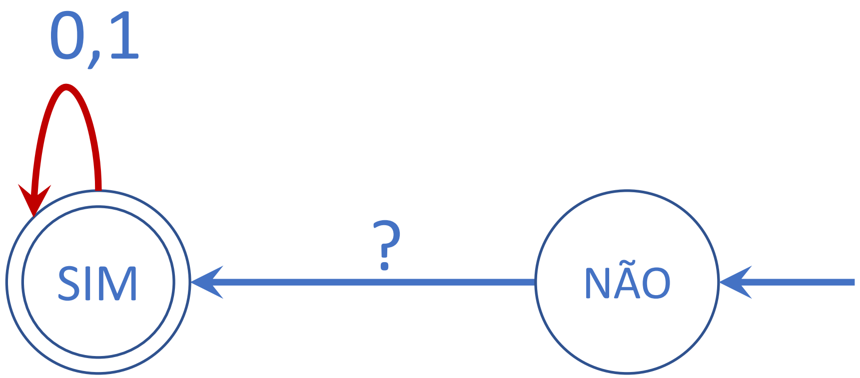
Desenvolvendo autômatos finitos

- É possível mudar de um desses dois estados para o outro? Com um só símbolo? Sempre ou depende do que aconteceu antes?



Desenvolvendo autômatos finitos

- Depois de visto o padrão, não pode voltar, não importa mais o resto da sequência

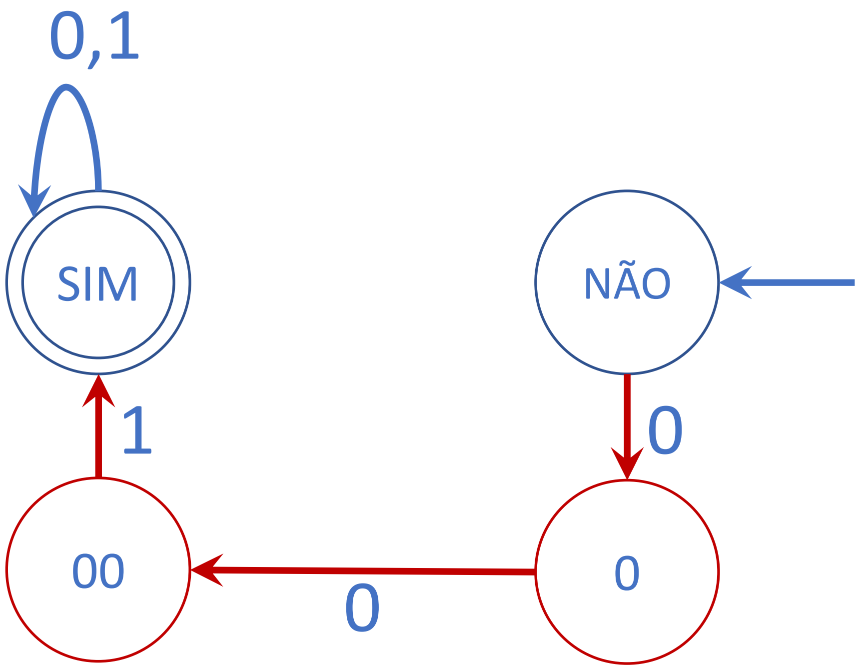


Desenvolvendo autômatos finitos

- Não posso mudar diretamente, na realidade existem estados intermediários
  - Não vi o padrão ainda...
    - Nem vi nenhum pedaço dele
    - Mas vi o que pode ser um pedaço dele
      - Vi um 0
      - Vi um 00

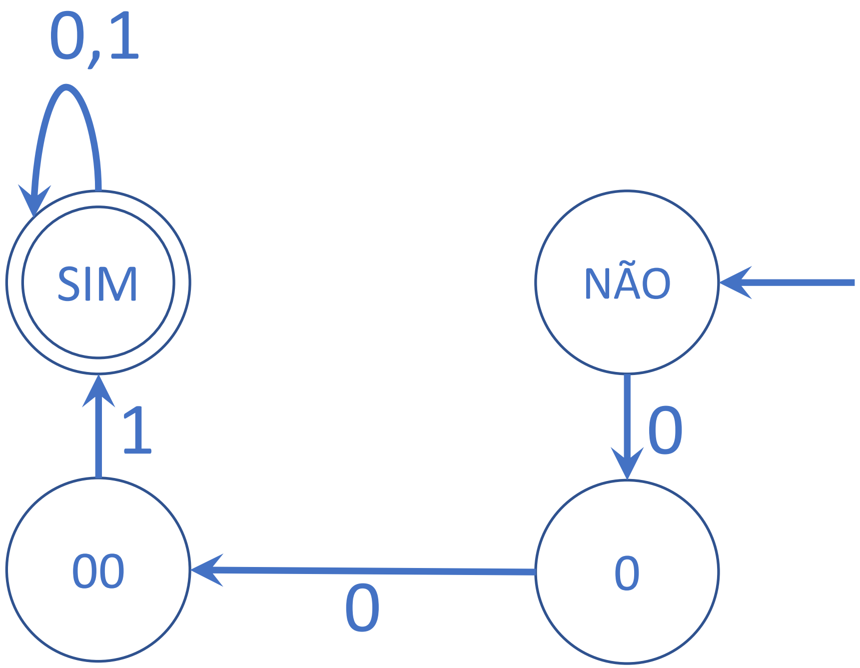
Desenvolvendo autômatos finitos

- Se não vi o padrão ainda... mas vi o que pode ser um pedaço dele
  - Vi “0”
  - Vi “00”



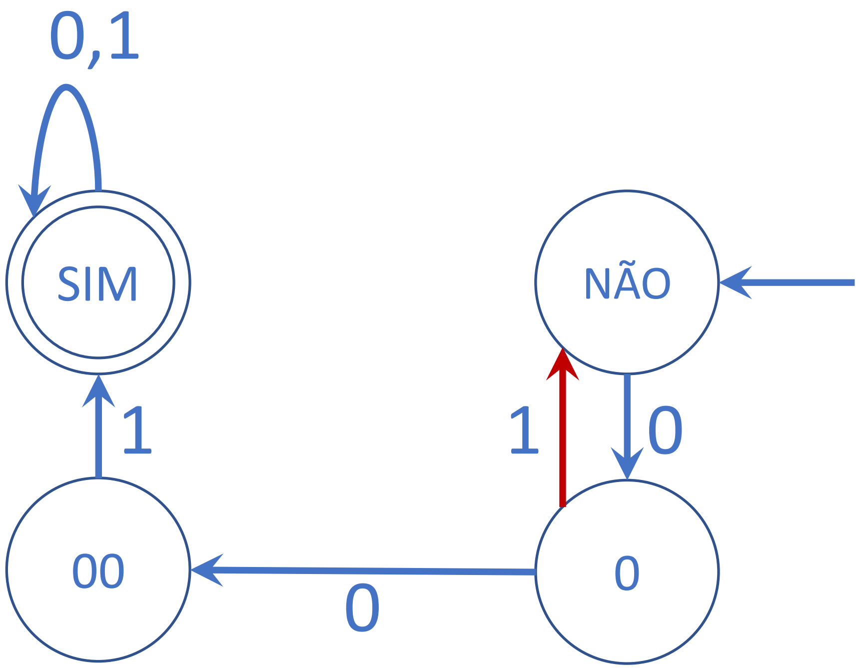
Desenvolvendo autômatos finitos

- Que outras possibilidades existem? Que arcos faltam? (Que buracos eu teria na tabela?)



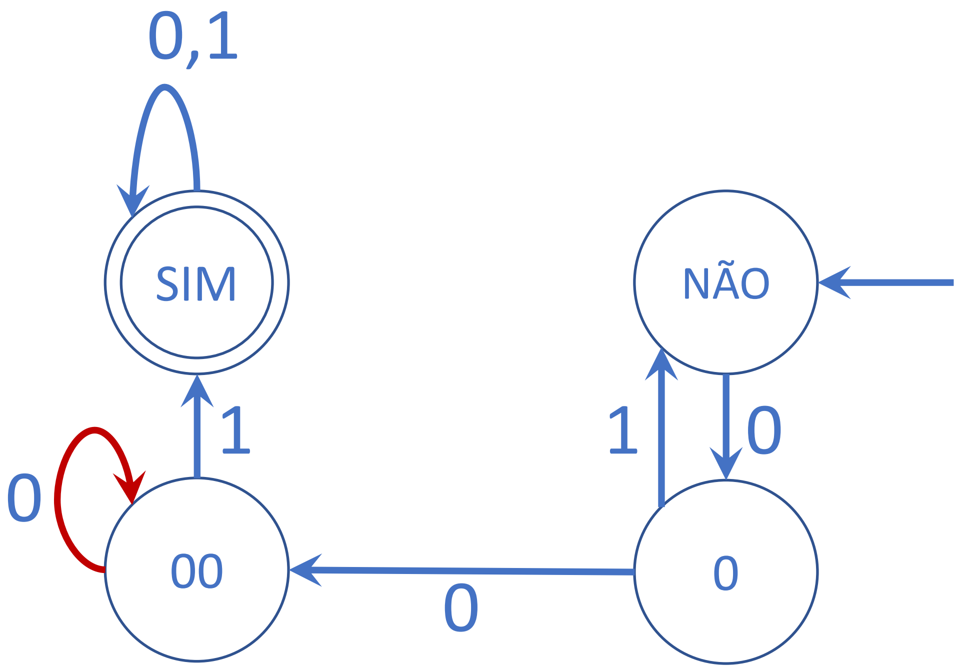
Desenvolvendo autômatos finitos

- Tinha visto um 0, mas não vi o segundo, ao invés disso vi um 1



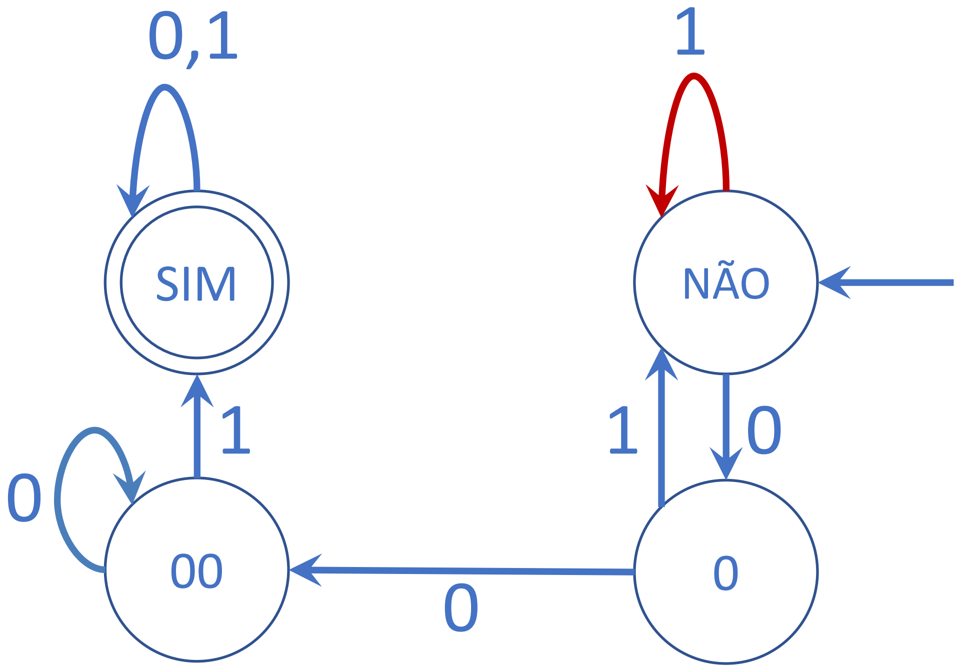
Desenvolvendo autômatos finitos

- Tinha visto 00, mas recebi um novo 0



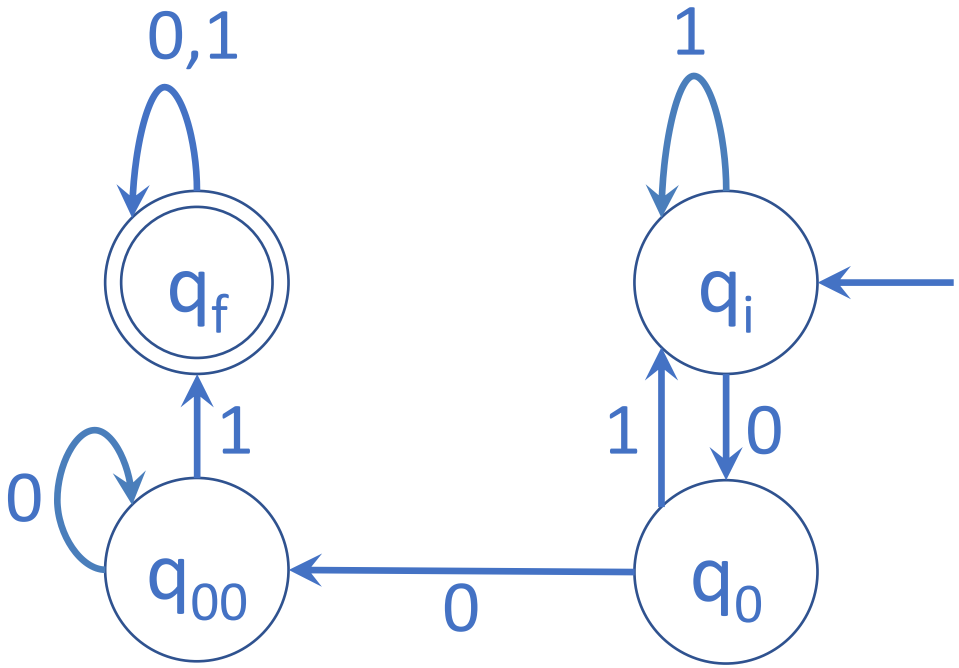
Desenvolvendo autômatos finitos

- Atualmente, não tenho nem uma parte do padrão



Desenvolvendo autômatos finitos

- Autômato final



Outros exemplos

- Um autômato que aceita palavras que tenham um número par de 0s
  - Jeito errado de resolver: contar o total de 0s
  - Jeito certo: lembrar apenas do mínimo de informação
- Um autômato que aceita palavras com um número par de 0s e par de 1s
  - Quantos estados tem esse AF?
- Um autômato que aceita palavras com um número par de 0s e de 1s **ou** par de 0s e ímpar de 1s
- Mais um: todo 1 é seguido por dois 0s

Outros exemplos

- Um autômato que aceita palavras que tenham um número par de 0s
  - Jeito errado de resolver: contar o total de 0s
  - Jeito certo: lembrar apenas do mínimo de informação

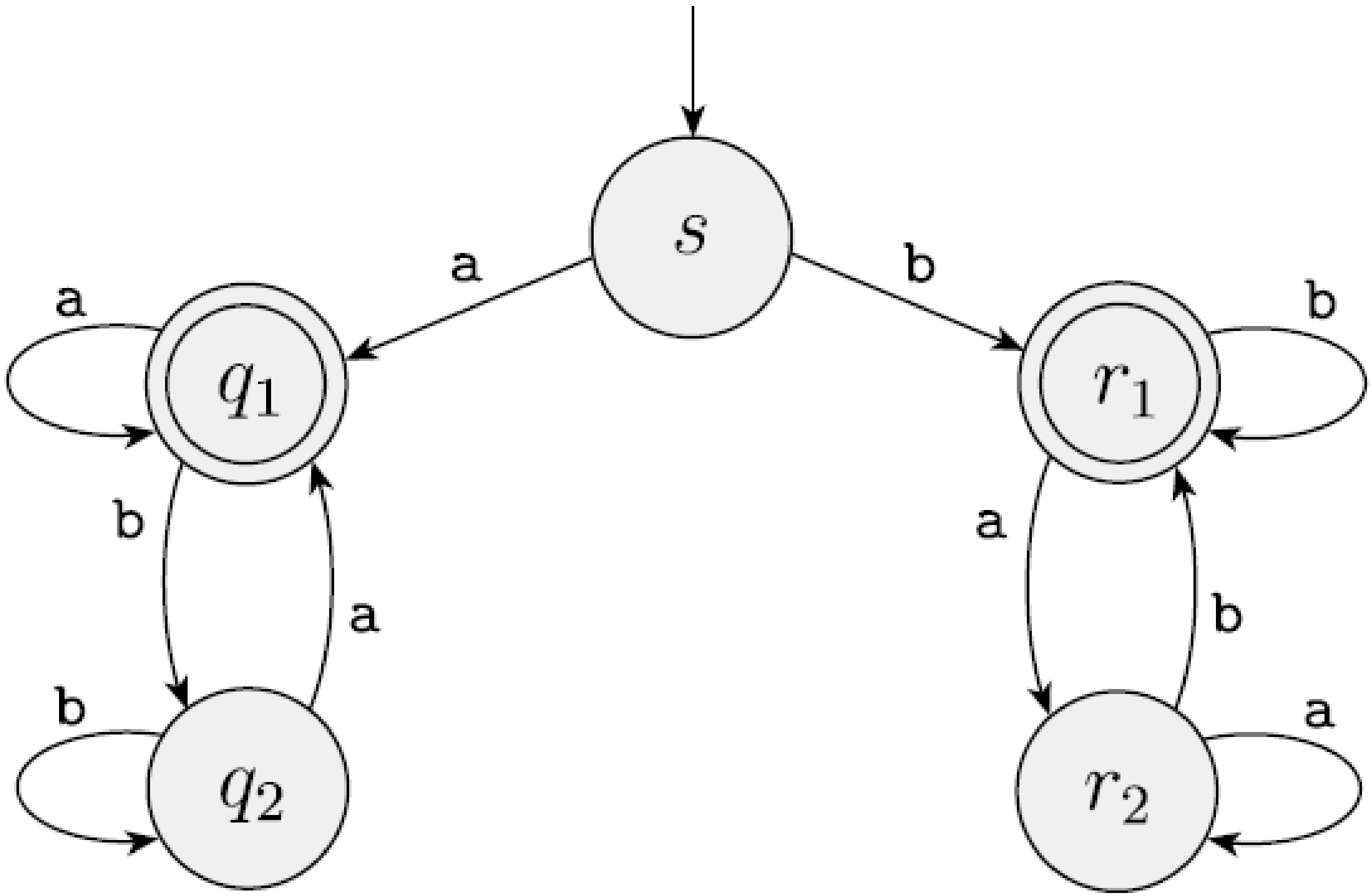
- Um autômato que aceita palavras com um número par de 0s e par de 1s
  - Quantos estados tem esse AF?

- Um autômato que aceita palavras com um número par de 0s e de 1s **ou** par de 0s e ímpar de 1s

- Mais um: todo 1 é seguido por dois 0s
  - (Pergunta: e o que acontece com as palavras sem 1?)

Exemplo: M_4

- $L(M_4)$?



Exemplo inverso

- Qual o diagrama de estados do autômato $<Q, \sum, \delta, q_0, F>$, onde:
  - $Q = \{q_0, q_1, q_2\}$
  - $\sum = \{R, 0, 1, 2\}$
  - $\delta: Q \times \sum \rightarrow Q$ é tal que
    - $\delta(q_j,0)=q_j$
    - $\delta(q_j,1)=q_k$, onde $k = (j+1)$ módulo 3
    - $\delta(q_j,2)=q_k$, onde $k = (j+2)$ módulo 3
    - $\delta(q_j,R)=q_0$
  - $F = \{q_0\}$

Classe de linguagens

- Um conjunto de linguagens que podem ser reconhecidas pelo mesmo tipo de máquina
  - No nosso caso, a máquina é o autômato finito

- O nome da classe que contém todas as linguagem que podem ser reconhecidas por AFs é a "classe das linguagem regulares"
  - Se existe um autômato que reconhece aquela linguagem, ela é regular
  - (E isso tem um significado sobre os recursos necessários para resolver aque;e problema)

Onde estamos?

- Precisamos saber olhar para um diagrama de estados de um AF e defini-lo formalmente, e vice-versa
- Construir autômatos finitos para um problema qualquer (??)
- Entender o vocabulário e a notação
  - Alfabeto, linguagem e cadeia
  - Função de transição, estados finais e iniciais
  - Aceitação e rejeição