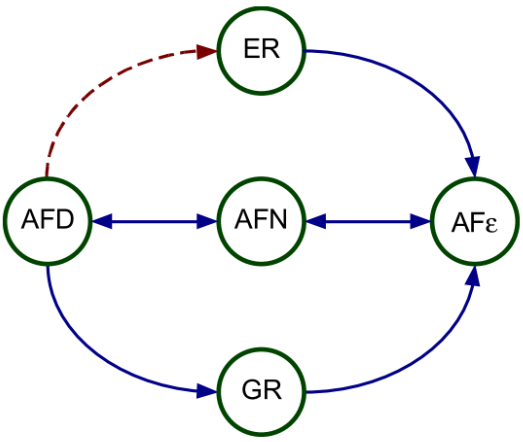
Propriedades das linguagens regulares

Características gerais

- Baixa complexidade
- Grande eficiência
- Fácil implementação

LIMITAÇÕES

- Classe restrita e limitada
- É fácil definir linguagens não regulares



Lema do Bombeamento para LR

Teorema: Se L é uma linguagem regular, então existe um inteiro n tal que para qualquer string w em L, onde o comprimento de w é maior ou igual a n $|w| \geq n$, pode-se dividir w em três partes, w = uvz, que satisfazem:

- $|uv| \leq n$
- $|v| \geq 1$
- para todo $i \geq 0$, uv^iz

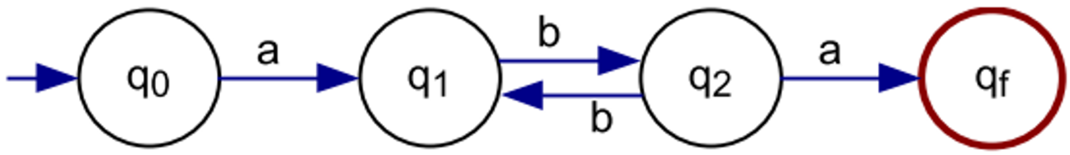
Prova: Baseada na existência de ciclos em um AFD devido ao número finito de estados.

APLICAÇÃO DO LEMA DO BOMBEAMENTO (EXEMPLO)

- Escolhemos $w = 0^n1^n$ que está em L
- Dividimos w em $|uv| \leq n$ $|v| \geq 1$

- Como |uv| ≤ n e w começa com n 0’s, sabemos que u e v consistem apenas de 0’s, portanto, v contém apenas 0’s
- Escolhendo i = 0, então uv^0z = uz
- Como v contém apenas 0’s, remover v de w resulta em uma string com menos 0’s que 1’s
- Remover v (que contém pelo menos 1 zero) resulta em uz com menos de n 0’s, mas ainda n 1’s
- Portanto, como uz não é palavra de L, L não é regular

(EXEMPLO)



- n = 4
- No caso particular de w = abbba
  - qr = qs = q1
  - u = a
  - v = bb
  - z = ba
- L é regular pelo teorema do bombeamento

Linguagem não regular

- Necessita ser desenvolvida para cada caso
- Algumas ferramentas específicas
  - Exemplo: Bombeamento e demostração por absurdo

Operações fechadas sobre LR

> para uma linguagem L sobre Σ*, L' denota o seu complemento em Σ*

- A classe das LR é fechada para: União, concatenação, complemento, intersecção

Prova do fechamento:

- União e concatenação: derivam da definição de ER
- Complemento: Inverter condições de aceitação/rejeição do AF

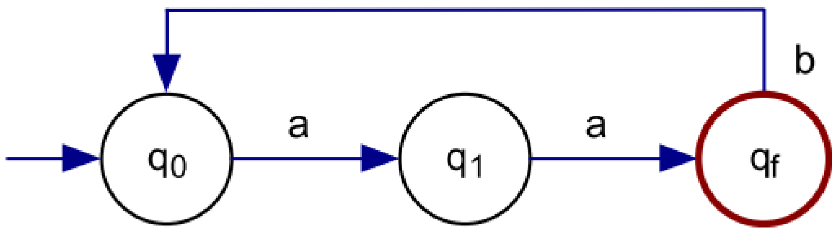
    - Transformar os estados finais em não finais e vice-versa
- Intersecção: Usando complemento e união
    - $L1 \cap L2 = (L1' \cup L2')'$

Investigação se uma LR é vazia, finita ou infinita

Teorema: Se L é LR aceita por um AF M com n estados, então L é:

- Vazia: sse M não aceita qualquer w tq $|w| < n$
- Finita: sse M não aceita w tq $n \leq |w| < 2n$
- Infinita: sse M aceita w tq n $n \leq |w| < 2n$
- Existe um algoritmo para verificar essas propriedades

EXEMPLO (LINGUAGEM INFINITA)



- Infinita sse M aceita w tq n $n \leq |w| < 2n$
- Número de estado = 3, então n = 3
- aabaa
    - Possui comprimento 5
    - Ou seja, $3 \leq |aabaa| < 6$
- Sendo assim, é infinita

Igualdade de linguagens regulares

Teorema: Se M1 e M2 são AF, existe um algoritmo para determinar se ACEITA(M1) = ACEITA(M2)

Prova:

- Contruir um AF M3 tal que $L(M3) = (L1 \cap L2') \cup (L1' \cap L2)$
- L1 = L2 sse L3 é vazia.
- Algoritmo para verificar se LR é vazia, interseção e complemento

Eficiência de um AF como algoritmo de reconhecimento

Simulação:

- Fácil implementação.
- Algoritmo que controla mudanças de estado a cada símbolo lido.

Tempo de processamento:

- Proporcional ao tamanho da entrada.
- Pertence à classe mais rápida de algoritmos.

Otimização:

- Algoritmo para construir AFD mínimo (menor número de estados).