Teoria da Computação

Prof. Maicon R. Zatelli

Aula 4 - Decidibilidade (Linguagens Decidíveis e Linguagens Indecidíveis) - Parte 1

Universidade Federal de Santa Catarina

Florianópolis - Brasil

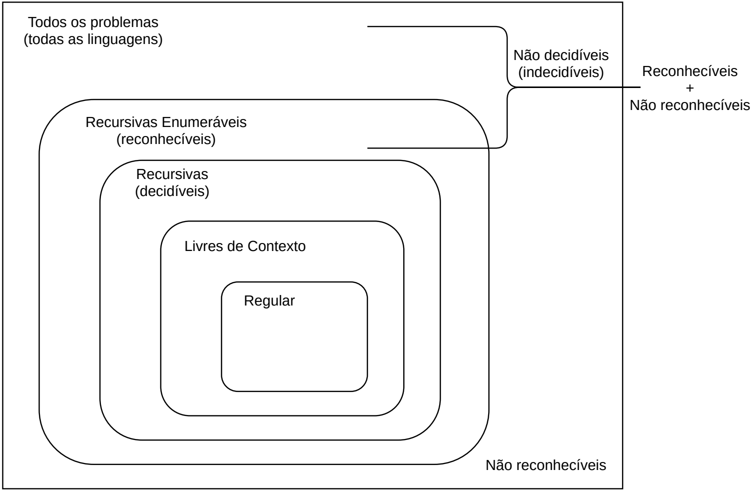
Introdução

- Linguagens decidíveis
- Método da diagonalização
- Linguagens indecidíveis

Material de apoio

- Livro Sipser, Capítulo 4
- Livro Hopcroft, Capítulo 9

Hierarquia de Chomsky



Indecidíveis - são todas as linguagens Turing-reconhecíveis mas não decidíveis e também as linguagens não Turing-reconhecíveis

Decidibilidade

Investigar o poder dos algoritmos

Demonstrar que certos problemas podem ser resolvidos algoritmicamente e outros não

Explorar os limites do poder dos algoritmos

- Problemas decidíveis
- Problemas indecidíveis

Linguagens Decidíveis

Em geral, mostramos que uma linguagem é decidível construindo uma MT que a decide

Linguagens Decidíveis

Em geral, mostramos que uma linguagem é decidível construindo uma MT que a decide

Vamos ver alguns exemplos com problemas envolvendo autômatos finitos

- Algoritmos para testar se um AFD aceita uma palavra, se a linguagem reconhecida por um AFD é vazia, se dois AFD são equivalentes, entre outros

Linguagens Decidíveis

Em geral, mostramos que uma linguagem é decidível construindo uma MT que a decide

Vamos ver alguns exemplos com problemas envolvendo autômatos finitos

- Algoritmos para testar se um AFD aceita uma palavra, se a linguagem reconhecida por um AFD é vazia, se dois AFD são equivalentes, entre outros

Primeiramente, representaremos os problemas por linguagens

Linguagens Decidíveis

Problema 1: Testar se um AFD aceita uma palavra w

Linguagens Decidíveis

Problema 1: Testar se um AFD aceita uma palavra w

Linguagem: $A_{DFA} = \{< B, w > | B$ é um AFD e B aceita a palavra $w\}$

Problema 1: Testar se um AFD aceita uma palavra w

Linguagem: $A_{DFA} = \{< B, w > | B$ é um AFD e B aceita a palavra $w\}$

Note que o problema de testar se B aceita a palavra w é o mesmo problema que testar se $< B, w >$ pertence à linguagem A_{DFA}

Problema 1: Testar se um AFD aceita uma palavra w

Linguagem: $A_{DFA} = \{< B, w > |B$ é um AFD e B aceita a palavra $w\}$

Note que o problema de testar se B aceita a palavra w é o mesmo problema que testar se $< B, w >$ pertence à linguagem A_{DFA}

Mostrar que a linguagem é decidível é o mesmo que mostrar que o problema computacional equivalente é decidível

Linguagens Decidíveis

Teorema 1: A_{DFA} é decidível

Linguagens Decidíveis

Teorema 1: A_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{DFA}

Linguagens Decidíveis

Teorema 1: A_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{DFA}

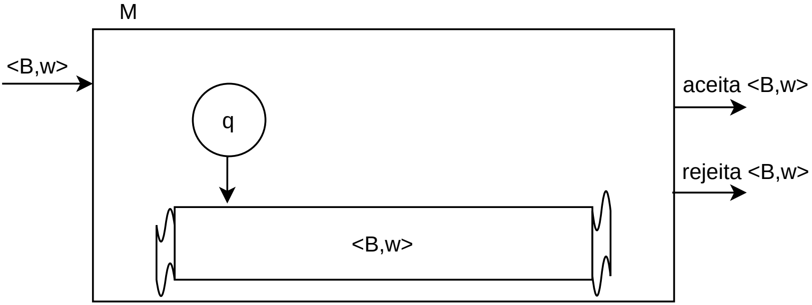
Seja M uma MT que decide A_{DFA}

```
M recebe como entrada <B,w>, onde B é a descrição do AFD e w é
uma palavra:
    1: Simule B com a entrada w
    2: Se a simulação terminar num estado de aceitação,
       aceite (M aceita <B,w>), caso contrário, rejeite
       (M rejeita <B,w>)
```

Teorema 1: A_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{DFA}

Seja M uma MT que decide A_{DFA}



Linguagens Decidíveis

Teorema 1: A_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{DFA}

- A fita da MT contém a descrição do AFD, juntamente com a palavra de entrada
- A MT mantém o estado atual do AFD e a posição apontada em w, escrevendo isso na fita
- Inicialmente, o estado atual é $q0$ e a posição em w é o primeiro símbolo de w
- São usados símbolos separadores de cada informação

Problema 2: Testar se um AFND aceita uma palavra w

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > |B$ é um AFND e B aceita a palavra $w\}$

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$

Teorema 2: A_{NFA} é decidível

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$

Teorema 2: A_{NFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{NFA}

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$

Teorema 2: A_{NFA} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{NFA}

Seja M uma MT que decide A_{NFA}

```
M com a entrada <B,w>, onde B é a descrição do AFND e w é uma
palavra, faz:
    1: Converta do AFND B em um AFD C
    2: Simule C com a entrada w
    3: Se C aceitar w, aceite (M aceita <B,w>), caso contrário,
       rejeite (M rejeita <B,w>)
```

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$

Teorema 2: A_{NFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide A_{NFA}

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$

Teorema 2: A_{NFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide A_{NFA}

Seja $M2$ uma MT que decide A_{NFA} e M uma MT que decide A_{DFA}

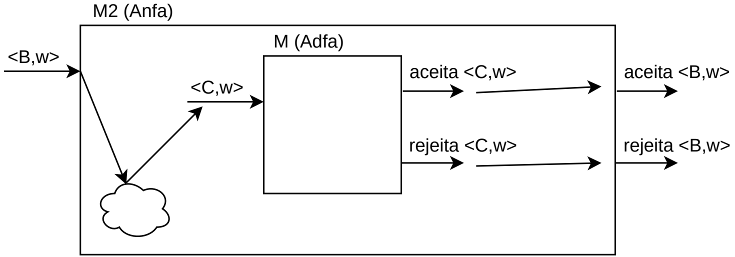
```
M2 com a entrada <B,w>, onde B é a descrição do AFND e w é uma
palavra, faz:
    1: Converta do AFND B em um AFD C
    2: Rode M (do Teorema 1) com a entrada <C,w>
    3: Se M aceitar <C,w>, aceite (M2 aceita <B,w>),
       caso contrário, rejeite (M2 rejeita <B,w>)
```

Problema 2: Testar se um AFND aceita uma palavra w

Linguagem: $A_{NFA} = \{< B, w > | B$ é um AFND e B aceita a palavra $w\}$
medskip
Teorema 2: A_{NFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide A_{NFA}

Seja $M2$ uma MT que decide A_{NFA} e M uma MT que decide A_{DFA}



Problema 3: Testar se uma expressão regular gera uma palavra w

Problema 3: Testar se uma expressão regular gera uma palavra w

Linguagem: $A_{REX} = \{< R, w > | R$ é uma ER e R gera a palavra $w\}$

Problema 3: Testar se uma expressão regular gera uma palavra w

Linguagem: $A_{REX} = \{< R, w > | R$ é uma ER e R gera a palavra $w\}$

Teorema 3: A_{REX} é decidível

Problema 3: Testar se uma expressão regular gera uma palavra w

Linguagem: $A_{REX} = \{< R, w > | R$ é uma ER e R gera a palavra $w\}$

Teorema 3: A_{REX} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{REX}

Problema 3: Testar se uma expressão regular gera uma palavra w

Linguagem: $A_{REX} = \{< R, w > | R$ é uma ER e R gera a palavra $w\}$

Teorema 3: A_{REX} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{REX}

Seja $M3$ uma MT que decide A_{REX} e M2 uma MT que decide A_{NFA}

```
M3 com a entrada <R,w>, onde R é uma ER e w é uma palavra, faz:
    1: Converta a ER R em um AFND D equivalente
    2: Rode M2 (do Teorema 2) com a entrada <D,w>
    3: Se M2 aceitar <D,w>, aceite (M3 aceita <R,w>),
       caso contrário, rejeite (M3 rejeita <R,w>)
```

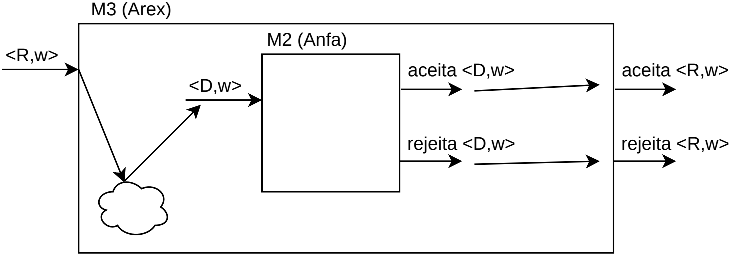
Problema 3: Testar se uma expressão regular gera uma palavra w

Linguagem: $A_{REX} = \{< R, w > | R$ é uma ER e R gera a palavra $w\}$

Teorema 3: A_{REX} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{REX}

Seja $M3$ uma MT que decide A_{REX} e M2 uma MT que decide A_{NFA}



Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A \text{ é um AFD e } L(A) = \emptyset\}$

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A \text{ é um AFD e } L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A \text{ é um AFD e } L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{DFA}

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A$ é um AFD e $L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
    1: Gera todas as palavras para cada tamanho t=0,1,2,...
       e simule A com cada palavra
    2: Se alguma palavra é aceita por A, rejeite (M4 rejeita <A>)
    3: Se nenhuma palavra é aceita por A, aceite (M4 aceita <A>)
```

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A \text{ é um AFD e } L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
    1: Gera todas as palavras para cada tamanho t=0,1,2,...
       e simule A com cada palavra
    2: Se alguma palavra é aceita por A, rejeite (M4 rejeita <A>)
    3: Se nenhuma palavra é aceita por A, aceite (M4 aceita <A>)
```

Problema?

Linguagens Decidíveis

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A$ é um AFD e $L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
    1: Gera todas as palavras para cada tamanho t=0,1,2,...
       e simule A com cada palavra
    2: Se alguma palavra é aceita por A, rejeite (M4 rejeita <A>)
    3: Se nenhuma palavra é aceita por A, aceite (M4 aceita <A>)
```

Problema? E quando é sabido que nenhuma palavra é aceita?

Linguagens Decidíveis

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A$ é um AFD e $L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
    1: Gera todas as palavras para cada tamanho t=0,1,2,...
       e simule A com cada palavra
    2: Se alguma palavra é aceita por A, rejeite (M4 rejeita <A>)
    3: Se nenhuma palavra é aceita por A, aceite (M4 aceita <A>)
```

Problema? E quando é sabido que nenhuma palavra é aceita?

- A MT testará palavras indefinidamente! Nunca irá parar.

Linguagens Decidíveis

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A$ é um AFD e $L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
  1: Marque o estado inicial
  2: Repita
    2.1: Para cada estado em A, marque-o se há alguma transição
         chegando a ele a partir de qualquer estado já marcado
    2.2: Se nenhum novo estado for marcado em 2.1, vá para 3
  3: Se nenhum estado de aceitação for marcado, aceite
     (M4 aceita <A>), caso contrário, rejeite (M4 rejeita <A>)
```

Linguagens Decidíveis

Problema 4: Testar se um AFD aceita a linguagem vazia $\emptyset$

Linguagem: $E_{DFA} = \{< A > | A \text{ é um AFD e } L(A) = \emptyset\}$

Teorema 4: E_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide E_{DFA}

Seja $M4$ uma MT que decide E_{DFA}

```
M4 com a entrada <A>, onde A é um AFD, faz:
  1: Marque o estado inicial
  2: Repita
    2.1: Para cada estado em A, marque-o se há alguma transição
         chegando a ele a partir de qualquer estado já marcado
    2.2: Se nenhum novo estado for marcado em 2.1, vá para 3
  3: Se nenhum estado de aceitação for marcado, aceite
     (M4 aceita <A>), caso contrário, rejeite (M4 rejeita <A>)
```

Note que agora a MT sempre irá parar, tanto para aceitar como para rejeitar

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide EQ_{DFA}

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide EQ_{DFA}

Seja $M5$ uma MT que decide EQ_{DFA}

```
M5 com a entrada <A,B>, onde A e B são AFDs, faz:
    1: Gera todas as palavras w para cada tamanho t=0,1,2,...
       e simule A e B com cada palavra w
    2: Se A e B divergem na resposta, rejeite (M5 rejeita <A,B>)
    3: Se A e B nunca divergem na resposta, aceite
       (M5 aceita <A,B>)
```

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide EQ_{DFA}

Seja $M5$ uma MT que decide EQ_{DFA}

```
M5 com a entrada <A,B>, onde A e B são AFDs, faz:
    1: Gera todas as palavras w para cada tamanho t=0,1,2,...
       e simule A e B com cada palavra w
    2: Se A e B divergem na resposta, rejeite (M5 rejeita <A,B>)
    3: Se A e B nunca divergem na resposta, aceite
       (M5 aceita <A,B>)
```

Problema?

Linguagens Decidíveis

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide EQ_{DFA}

Seja $M5$ uma MT que decide EQ_{DFA}

```
M5 com a entrada <A,B>, onde A e B são AFDs, faz:
    1: Gera todas as palavras w para cada tamanho t=0,1,2,...
       e simule A e B com cada palavra w
    2: Se A e B divergem na resposta, rejeite (M5 rejeita <A,B>)
    3: Se A e B nunca divergem na resposta, aceite
       (M5 aceita <A,B>)
```

Problema? E quando é sabido que A e B nunca divergem?

Linguagens Decidíveis

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova: prova por construção. Construimos uma MT que decide EQ_{DFA}

Seja $M5$ uma MT que decide EQ_{DFA}

```
M5 com a entrada <A,B>, onde A e B são AFDs, faz:
    1: Gera todas as palavras w para cada tamanho t=0,1,2,...
       e simule A e B com cada palavra w
    2: Se A e B divergem na resposta, rejeite (M5 rejeita <A,B>)
    3: Se A e B nunca divergem na resposta, aceite
       (M5 aceita <A,B>)
```

Problema? E quando é sabido que A e B nunca divergem?

- A MT testará palavras indefinidamente! Nunca irá parar.

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > |A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Vale a pena lembrar de que as LR são fechadas nas operações de complemento, união, intersecção, diferença, entre outras

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Vale a pena lembrar de que as LR são fechadas nas operações de complemento, união, intersecção, diferença, entre outras

O que o problema pede é se não há palavras apenas aceitas ou por A ou por B

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

A linguagem das palavras aceitas por A, mas não aceitas por B é dada por $L_C = L(A) \cap \overline{L(B)}$

Linguagens Decidíveis

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

A linguagem das palavras aceitas por A, mas não aceitas por B é dada por $L_C = L(A) \cap \overline{L(B)}$

A linguagem das palavras aceitas por B, mas não aceitas por A é dada por $L_D = \overline{L(A)} \cap L(B)$

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

A linguagem das palavras aceitas por A, mas não aceitas por B é dada por $L_C = L(A) \cap \overline{L(B)}$

A linguagem das palavras aceitas por B, mas não aceitas por A é dada por $L_D = \overline{L(A)} \cap L(B)$

A linguagem das palavras aceitas ou só por A ou só por B é dada por $L_E = L_C \cup L_D$

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > |A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Agora basta verificar se L_E é vazia ou não! Se L_E é vazia significa que não há palavras reconhecidas por apenas ou A ou B

Linguagens Decidíveis

Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Agora basta verificar se L_E é vazia ou não! Se L_E é vazia significa que não há palavras reconhecidas por apenas ou A ou B

Seja $M5$ uma MT que decide EQ_{DFA}

```
M5 com a entrada <A,B>, onde A e B são AFDs, faz:
    1: Construa um AFD J que reconheça a linguagem L_E
    2: Rode M4 (do teorema 4) com a entrada <J>
    3: Se M4 aceita <J>, aceite (M5 aceita <A,B>), caso contrário,
       rejeite (M5 rejeita <A,B>)
```

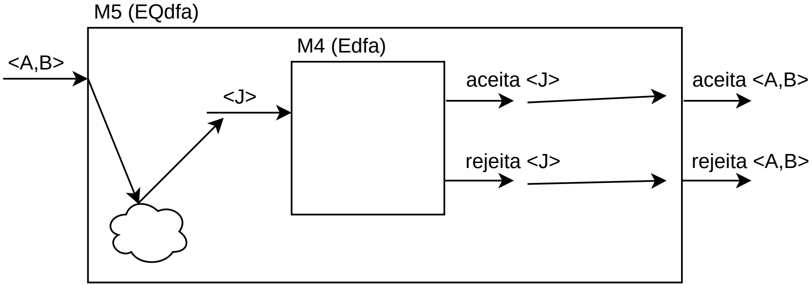
Problema 5: Testar se dois AFDs aceitam a mesma linguagem

Linguagem: $EQ_{DFA} = \{< A, B > | A$ e B são AFDs e $L(A) = L(B)\}$

Teorema 5: EQ_{DFA} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide EQ_{DFA}

Seja $M5$ uma MT que decide EQ_{DFA}



Linguagens Decidíveis

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

Linguagens Decidíveis

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

- $M6$ produz derivações e testa se alguma derivação gerou a palavra w

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

- $M6$ produz derivações e testa se alguma derivação gerou a palavra w

Problema?

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

- $M6$ produz derivações e testa se alguma derivação gerou a palavra w

Problema? E se w não pertence a linguagem gerada pela GLC G?

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

- $M6$ produz derivações e testa se alguma derivação gerou a palavra w

Problema? E se w não pertence a linguagem gerada pela GLC G?

- A MT testará derivações indefinidamente! Nunca irá parar.

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

- $M6$ produz derivações e testa se alguma derivação gerou a palavra w

Problema? E se w não pertence a linguagem gerada pela GLC G?

- A MT testará derivações indefinidamente! Nunca irá parar.
- Solução: Precisamos testar um número limitado de derivações!

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Base da indução: $|w| = 1$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Base da indução: $|w| = 1$

- Neste caso, w possui um único símbolo a

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Base da indução: $|w| = 1$

- Neste caso, w possui um único símbolo a
- Qualquer derivação de w será com exato 1 passo, usando uma regra na forma $X \rightarrow a$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Base da indução: $|w| = 1$

- Neste caso, w possui um único símbolo a
- Qualquer derivação de w será com exato 1 passo, usando uma regra na forma $X \rightarrow a$
- Assim, temos que $2 * 1 - 1 = 1$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Se $k = 1$, estamos no caso base

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Se $k = 1$, estamos no caso base
- Senão, o primeiro passo da derivação irá usar uma regra da forma $X \rightarrow AB$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Se $k = 1$, estamos no caso base
- Senão, o primeiro passo da derivação irá usar uma regra da forma $X \rightarrow AB$
- Os próximos passos de derivação irão expandir A até alguma palavra wA e B até alguma palavra wB, sendo que nem wA e nem wB é a palavra vazia

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Seja então $|wA| = i$ e $|wB| = j$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Seja então $|wA| = i$ e $|wB| = j$
- Portanto, temos que $|wAwB| = i + j = k$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Seja então $|wA| = i$ e $|wB| = j$
- Portanto, temos que $|wAwB| = i + j = k$
- Como tanto wA como wB possuem algum símbolo, então $|wA| > 0$, $|wB| > 0$ e então $i < k$ e $j < k$

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Seja então $|wA| = i$ e $|wB| = j$
- Portanto, temos que $|wAwB| = i + j = k$
- Como tanto wA como wB possuem algum símbolo, então $|wA| > 0$, $|wB| > 0$ e então $i < k$ e $j < k$
- Pela nossa hipótese de indução, temos que:

Linguagens Decidíveis

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Seja então $|wA| = i$ e $|wB| = j$
- Portanto, temos que $|wAwB| = i + j = k$
- Como tanto wA como wB possuem algum símbolo, então $|wA| > 0$, $|wB| > 0$ e então $i < k$ e $j < k$
- Pela nossa hipótese de indução, temos que:
  - wA será derivada em $2i - 1$ passos
  - wB será derivada em $2j - 1$ passos

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Note que w pode então ser derivada a partir de X em:
  - **1** ($X \to AB$) + **(2i - 1)** (deriva wA) + **(2j-1)** (deriva wB) **passos**

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Note que w pode então ser derivada a partir de X em:
  - **1** ($X \rightarrow AB$) **+ (2i - 1)** (deriva wA) **+ (2j-1)** (deriva wB) **passos**
- Assim, veja que
  - $1 + (2i - 1) + (2j - 1) = 1 + 2i + 2j - 1 - 1 = 2(i + j) - 1$

Linguagens Decidíveis

Teorema 7: Se G é uma GLC na forma normal de Chomsky, então qualquer palavra $w \in L(G)$, com $|w| = n$ e $n \geq 1$, pode ser derivada em exatamente $2n - 1$ passos

Prova: prova por indução no tamanho de w

Passo indutivo: assuma que o teorema é válido para todos os tamanhos de palavras menores que k, ou seja, $|w| < k$

Vamos mostrar que o teorema é também válido para palavras de tamanho k

- Note que w pode então ser derivada a partir de X em:
  - **1** ($X \rightarrow AB$) **+ (2i - 1)** (deriva wA) **+ (2j-1)** (deriva wB) **passos**
- Assim, veja que
  - $1 + (2i - 1) + (2j - 1) = 1 + 2i + 2j - 1 - 1 = 2(i + j) - 1$
  - $2(i + j) - 1 = 2k - 1$, onde $k = i + j$

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide A_{CFG}

Linguagens Decidíveis

Problema 6: Testar se uma gramática livre de contexto gera determinada palavra w

Linguagem: $A_{CFG} = \{< G, w > | G$ é uma gramática livre de contexto e gera a palavra $w\}$

Teorema 6: A_{CFG} é decidível

Prova (versão 2): prova por construção. Construimos uma MT que decide A_{CFG}

Seja $M6$ uma MT que decide A_{CFG}

```
M6 com a entrada <G,w>, onde G é uma GLC e w uma palavra, faz:
    1: Converta G para a forma normal de Chomsky
    2: Liste todas as derivações com 2n-1 passos, onde n = |w|.
       Se n = 0, liste todas as derivações com um passo.
    3: Se alguma derivação gerou w, aceite (M6 aceita <G,w>),
       caso contrário, rejeite (M6 rejeita <G,w>)
```

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Linguagem: $E_{CFG} = \{< G > | G$ é uma gramática livre de contexto e $L(G) = \emptyset\}$

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Linguagem: $E_{CFG} = \{< G > | G$ é uma gramática livre de contexto e $L(G) = \emptyset\}$

Teorema 8: E_{CFG} é decidível

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Linguagem: $E_{CFG} = \{< G > | G$ é uma gramática livre de contexto e $L(G) = \emptyset\}$

Teorema 8: E_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{CFG}

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Linguagem: $E_{CFG} = \{< G > | G$ é uma gramática livre de contexto e $L(G) = \emptyset\}$

Teorema 8: E_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{CFG}

A ideia é verificar se o símbolo inicial da gramática pode gerar ao menos uma palavra, ou seja, se ele não é infértil/improdutivo

Problema 8: Testar se uma gramática livre de contexto gera a linguagem vazia

Linguagem: $E_{CFG} = \{< G > | G$ é uma gramática livre de contexto e $L(G) = \emptyset\}$

Teorema 8: E_{CFG} é decidível

Prova: prova por construção. Construimos uma MT que decide E_{CFG}

Seja $M8$ uma MT que decide E_{CFG}

```
M8 com a entrada <G>, onde G é uma GLC, faz:
  1: Marque todos os símbolos terminais
  2: Repita até que nenhuma nova variável seja marcada
    2.1: Marque qualquer variável A em que G possui uma regra
         A→ V1V2...Vn e cada V1...Vn tenha sido marcado
         (seja Vi terminal ou não)
  3: Se o símbolo inicial não foi marcado, aceite (M8 aceita <G>),
     caso contrário, rejeite (M8 rejeita <G>)
```

Tente utilizar a prova do teorema 8 anterior para verificar se a gramática a seguir gera a linguagem vazia

$$S \rightarrow ABE$$
$$A \rightarrow CaB|D$$
$$B \rightarrow DC$$
$$C \rightarrow cd$$
$$D \rightarrow d|e$$
$$E \rightarrow EC$$

Conclusão

- Linguagens decidíveis
- Método da diagonalização
- Linguagens indecidíveis

Material de apoio

- Livro Sipser, Capítulo 4
- Livro Hopcroft, Capítulo 9