Teoria da Computação
Decidibilidade

Leonardo Takuno
{leonardo.takuno@gmail.com}

Centro Universitário Senac

Sumário

Sumário

Decidibilidade

Estamos prontos para atacar a questão:

> O que os computadores podem ou não fazer ?

Fazemos isso por considerar as questões:

> Quais linguagens são Turing decidíveis, Turing reconhecíveis, ou nenhum ?

Assumindo a tese de Church-Turing, essas são propriedades fundamentais das linguagens.

Sumário

Decidibilidade

- Para mostrar que uma linguagem é decidível:
  - Escreva um decisor que a decide:
  - Deve-se mostrar que o decisor:
    - Pára sobre todas as entradas;
    - Aceita *w* se e somente se *w* pertence à linguagem;

Decidibilidade

Vamos estudar algum **padrão** de máquinas que são decisores.
Estas máquinas nos ajudarão a construir provas mais complicadas.

Sumário

Decidibilidade

Teorema 4.1: A linguagem

$$A_{AFD} = \{\langle B, w\rangle | \text{B é um AFD que aceita a cadeia de entrada } w\}$$

é decidível.

Prova: Construimos um decisor M_{AFD} para A_{AFD}.
M_{AFD} = "Sobre a entrada $\langle B, w\rangle$, onde B é um AFD, e w, uma cadeia:

1. Simule B sobre a entrada w;
2. Se a simulação termina em um estado de aceitação, **aceite**. Se ela termina em um estado de não-aceitação, **rejeite**"

Decidibilidade

Teorema 4.2: A linguagem

$$A_{AFN} = \{\langle B, w\rangle | \text{B é um AFN que aceita a cadeia de entrada } w\}$$

é decidível.

Prova: Construimos um decisor N_{AFN} para A_{AFN}.
N_{AFN} = "Sobre a entrada $\langle B, w\rangle$ onde B é um AFN, e w, uma cadeia:

1. Converta AFN B para um AFD equivalente C, usando um procedimento para essa conversão.
2. Rode a MT M_{AFD} sobre a entrada $\langle C, w\rangle$.
3. Se M_{AFD} aceita, *aceite*; caso contrário, *rejeite*".

Decidibilidade

Teorema 4.3: A linguagem

$$A_{EXR} = \{\langle R, w\rangle | R \text{ é uma expressão regular que gera a cadeia } w\}$$

é decidível.

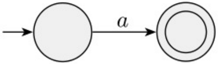
Ideia da prova:

- Converta a expressão regular R em AFN A e após isso simule o AFD A sobre w.
- Aceite se a simulação aceita, caso contrário rejeite.

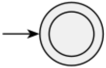
Decidibilidade

Convertendo a expressão regular R em AFN A:

$R = a$

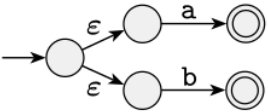


$R = \varepsilon$

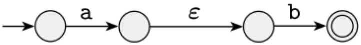


Decidibilidade

$R = a \cup b$

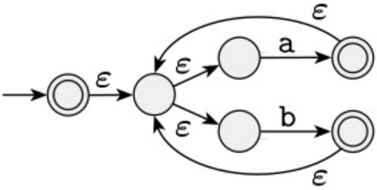


$R = a \cdot b$



Decidibilidade

$R = (a \cup b)^*$



Decidibilidade

Teorema 4.3: A linguagem

$$A_{EXR} = \{\langle R, w\rangle | R \text{ é uma expressão regular que gera a cadeia } w\}$$

é decidível.

Prova: Construimos um decisor P_{EXR} para A_{EXR}.
P_{EXR} = "Sobre a entrada $\langle R, w\rangle$, onde R é uma expressão regular e w é uma cadeia:

1. Converta a expressão regular R para um AFN equivalente A usando um procedimento para essa conversão.
2. Rode a MT N_{AFN} sobre a entrada $\langle A, w\rangle$
3. Se N_{AFN} aceita, *aceite*, caso contrário, *rejeite*"

Decidibilidade

Teorema 4.4: A linguagem

$$V_{AFD} = \{\langle A\rangle | A \text{ é um AFD e L(A)} = \emptyset\}$$

é decidível.

Prova: Construimos um decisor T para V_{AFD}.

T = "Sobre a entrada $\langle A\rangle$ onde A é um AFD:

1. Marque o estado inicial de A;
2. Repita até que nenhum estado novo venha a ser marcado.
3. Marque qualquer estado que tenha uma transição chegando nele a partir de qualquer estado que já está marcado.
4. Se nenhum estado de aceitação estiver marcado, *aceite*, caso contrário *rejeite*.

Decidibilidade

Teorema 4.5: A linguagem

$$EQ_{AFD} = \{\langle A, B\rangle |\ \text{A e B são AFDs e L(A) = L(B)}\}$$

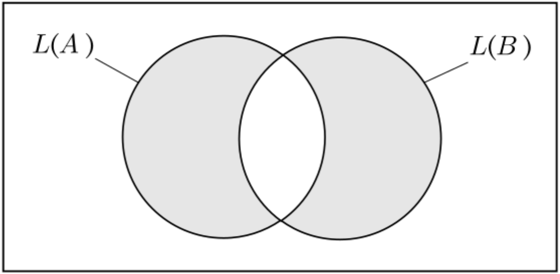
é decidível.

Prova: Para provar esse teorema construímos um novo AFD C que aceita somente as cadeias que são aceitas ou por A ou por B, mas não por ambos. Isto é,

$$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)). \tag{1}$$

Além disso, $\mathrm{L(C)} = \emptyset$ implica $\mathrm{L(A)} = \mathrm{L(B)}$. A classe das linguagens regulares é fechada sob complemento, união e intersecção. Portanto é possível construir uma máquina C de acordo com 1

Decidibilidade



Diferença simétrica de $L(A)$ e $L(B)$.

Decidibilidade

Podemos construir um decisor F para EQ_{AFD}.
$F =$ "Sobre a entrada $\langle A, B\rangle$ onde A e B são AFDs:

1. Construa o AFD C conforme descrito em 1.
2. Rode a MT T do Teorema anterior sobre a entrada $\langle C\rangle$.
3. Se T aceita, *aceite*. Se T rejeita, *rejeite*"

Sumário

Decidibilidade

Teorema 4.7: A linguagem

$$A_{GLC} = \{\langle G, w\rangle |\ \text{G é uma GLC que gera a cadeia}\ w\}$$

é decidível.

Ideia da prova:

- Dado GLC G e cadeia w, queremos determinar se G gera w.
- Para construir um decisor para A_{GLC}, precisamos garantir que o algoritmo tenta uma quantidade finita de derivações.
- Para isso faremos uma transformação de G para forma normal de Chomsky.

Decidibilidade

Definição: Uma gramática está na forma normal de Chomsky (FNC) se todas as suas regras de produção são da forma:

$$A \rightarrow BC \text{ ou}$$
$$A \rightarrow \alpha$$

onde A, B e C são variáveis e α é um símbolo terminal.

Qualquer derivação de w na FNC tem $2n - 1$ passos, onde n é o comprimento de w.

Derivação usando GLC na FNC

Em cada passo, o tamanho cresce em exatamente 1.

- $S \rightarrow SS$
- $S \rightarrow a$

Como exemplo, faremos derivações sucessivas para gerar a palavra $w = aaaaa$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

S

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$S \underset{1}{\Rightarrow} SS$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$

Usamos (n-1) passos até o momento. Agora, utilizaremos 1 passo adicional para cada terminal.

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS \underset{2}{\Rightarrow} aaSSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS \underset{2}{\Rightarrow} aaSSS \underset{3}{\Rightarrow} aaaSS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS \underset{2}{\Rightarrow} aaSSS \underset{3}{\Rightarrow} aaaSS \underset{4}{\Rightarrow} aaaaS$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS \underset{2}{\Rightarrow} aaSSS \underset{3}{\Rightarrow} aaaSS \underset{4}{\Rightarrow} aaaaS \underset{5}{\Rightarrow} aaaaa$$

Derivação usando GLC na FNC

- $S \rightarrow SS$
- $S \rightarrow a$

$w = aaaaa$, $n = 5$, onde n é o tamanho de w.

$$S \underset{1}{\Rightarrow} SS \underset{2}{\Rightarrow} SSS \underset{3}{\Rightarrow} SSSS \underset{4}{\Rightarrow} SSSSS$$
$$\underset{1}{\Rightarrow} aSSSS \underset{2}{\Rightarrow} aaSSS \underset{3}{\Rightarrow} aaaSS \underset{4}{\Rightarrow} aaaaS \underset{5}{\Rightarrow} aaaaa$$

$\therefore$ Toda derivação tem exatamente $2n - 1$ passos.

Decidibilidade

Teorema 4.7: A linguagem

$$A_{GLC} = \{\langle G, w\rangle |\ \text{G é uma GLC que gera a cadeia}\ w\}$$

é decidível.

Prova : Constuir a MT S para A_{GLC} :

S = "Sobre $\langle G, w\rangle$, onde G é uma GLC, e w, uma cadeia:

1. Converta G para forma normal de Chomsky
2. Liste todas as derivações com $2n - 1$ passos, onde n é o comprimento de w, exceto se $n = 0$; nesse último caso, liste todas as derivações com 1 passo.
3. Se alguma dessas derivações gera w, *aceite*; se não, *rejeite*"

Decidibilidade

Teorema 4.8: A linguagem

$$V_{GLC} = \{\langle G \rangle | \text{G é uma GLC e } L(G) = \emptyset\}$$

é decidível.

Ideia da prova:

- Para determinar se a linguagem de uma gramática é vazia, precisamos testar se a variável inicial pode gerar cadeia de terminais.
- Comece por marcar símbolos terminais, e depois faça uma varredura em todas as regras gramaticais.

Decidibilidade

- Se houver uma regra que permita alguma variável ser substituída por uma cadeia de símbolos marcados, então marque essa variável.
- O algoritmo continua dessa forma até que não possa marcar mais nenhuma variável.
- Caso a variável inicial não esteja marcada, então $L(G) = \emptyset$

Decidibilidade

$$S \to ABCD$$
$$A \to BCA$$
$$A \to xyz$$
$$B \to CA$$
$$B \to AB$$
$$C \to CB$$
$$C \to ww$$
$$D \to DD$$
$$D \to BD$$
$$D \to DC$$

Quais variáveis podem gerar uma string de terminais?

Decidibilidade

$$
\begin{aligned}
S &\rightarrow ABCD \\
A &\rightarrow BCA \\
A &\rightarrow xyz \\
B &\rightarrow CA \\
B &\rightarrow AB \\
C &\rightarrow CB \\
C &\rightarrow ww \\
D &\rightarrow DD \\
D &\rightarrow BD \\
D &\rightarrow DC
\end{aligned}
$$

Inicialmente marque os terminais.

Decidibilidade

$$
\begin{aligned}
S &\rightarrow ABCD \\
A &\rightarrow BCA \\
A &\rightarrow \underline{xyz} \\
B &\rightarrow CA \\
B &\rightarrow AB \\
C &\rightarrow CB \\
C &\rightarrow \underline{ww} \\
D &\rightarrow DD \\
D &\rightarrow BD \\
D &\rightarrow DC
\end{aligned}
$$

Uma regra pode ser substituída por uma cadeia de símbolos marcados?

Decidibilidade

$$
\begin{aligned}
S &\rightarrow ABCD \\
A &\rightarrow BCA \\
A &\rightarrow \underline{xyz} \\
B &\rightarrow CA \\
B &\rightarrow AB \\
C &\rightarrow CB \\
C &\rightarrow \underline{ww} \\
D &\rightarrow DD \\
D &\rightarrow BD \\
D &\rightarrow DC
\end{aligned}
$$

Marque essa variável.

Decidibilidade

$$
\begin{aligned}
S &\rightarrow \underline{A}B\underline{C}D \\
A &\rightarrow B\underline{C}\underline{A} \\
\underline{A} &\rightarrow \underline{xyz} \\
B &\rightarrow \underline{C}\underline{A} \\
B &\rightarrow \underline{A}B \\
C &\rightarrow \underline{C}B \\
\underline{C} &\rightarrow \underline{ww} \\
D &\rightarrow DD \\
D &\rightarrow BD \\
D &\rightarrow D\underline{C}
\end{aligned}
$$

Marque essa variável.

Decidibilidade

$$S \rightarrow \underline{A}B\underline{C}D$$
$$A \rightarrow B\underline{CA}$$
$$\underline{A} \rightarrow \underline{xyz}$$
$$B \rightarrow \underline{CA}$$
$$B \rightarrow \underline{A}B$$
$$C \rightarrow \underline{C}B$$
$$\underline{C} \rightarrow \underline{ww}$$
$$D \rightarrow DD$$
$$D \rightarrow BD$$
$$D \rightarrow D\underline{C}$$

Existe alguma outra regra que pode ser substituída por uma cadeia de símbolos marcados.

Decidibilidade

$$
\begin{aligned}
S &\rightarrow \underline{A}B\underline{C}D \\
A &\rightarrow B\underline{C}\underline{A} \\
\underline{A} &\rightarrow \underline{xyz} \\
B &\rightarrow \underline{CA} \\
B &\rightarrow \underline{A}B \\
C &\rightarrow \underline{C}B \\
\underline{C} &\rightarrow \underline{ww} \\
D &\rightarrow DD \\
D &\rightarrow BD \\
D &\rightarrow D\underline{C}
\end{aligned}
$$

Marque essa variável.

Decidibilidade

$$
\begin{aligned}
S &\to \underline{ABC}D \\
\underline{A} &\to \underline{BCA} \\
\underline{A} &\to \underline{xyz} \\
\underline{B} &\to \underline{CA} \\
\underline{B} &\to \underline{AB} \\
\underline{C} &\to \underline{CB} \\
\underline{C} &\to \underline{ww} \\
D &\to DD \\
D &\to \underline{B}D \\
D &\to D\underline{C}
\end{aligned}
$$

Marque essa variável.

Decidibilidade

$$S \rightarrow \underline{ABC}D$$
$$\underline{A} \rightarrow \underline{BCA}$$
$$\underline{A} \rightarrow \underline{xyz}$$
$$\underline{B} \rightarrow \underline{CA}$$
$$\underline{B} \rightarrow \underline{AB}$$
$$\underline{C} \rightarrow \underline{CB}$$
$$\underline{C} \rightarrow \underline{ww}$$
$$D \rightarrow DD$$
$$D \rightarrow \underline{B}D$$
$$D \rightarrow D\underline{C}$$

Existe alguma outra regra que pode ser substituída por uma cadeia de símbolos marcados.

Decidibilidade

$$
\begin{aligned}
S &\rightarrow \underline{ABC}D \\
\underline{A} &\rightarrow \underline{BCA} \\
\underline{A} &\rightarrow \underline{xyz} \\
\underline{B} &\rightarrow \underline{CA} \\
\underline{B} &\rightarrow \underline{AB} \\
\underline{C} &\rightarrow \underline{CB} \\
\underline{C} &\rightarrow \underline{ww} \\
D &\rightarrow DD \\
D &\rightarrow \underline{B}D \\
D &\rightarrow D\underline{C}
\end{aligned}
$$

Não existe, portanto, a variável inicial não pode ser marcada.

Decidibilidade

Teorema 4.8: A linguagem

$$V_{GLC} = \{\langle G \rangle | \text{G é uma GLC e } L(G) = \emptyset\}$$

é decidível.

Prova: Construa uma MT R para V_{GLC}

R = "Sobre a entrada $\langle G \rangle$, onde G é uma GLC:

1. Marque todos os símbolos terminais em G;
2. Repita até que nenhuma variável venha a ser marcada;
3. Marque qualquer variável A onde G tem uma regra $A \rightarrow U_1 U_2 \cdots U_k$ e cada símbolo $U_1 U_2 \cdots U_k$ já tenha sido marcado.
4. Se a variável inicial não está marcada, *aceite*, caso contrário, *rejeite*.

Igualdade de GLCs

E sobre a a igualdade das linguagens

$$EQ_{GLC} = \{\langle G, H\rangle | \text{G e H são GLCs com } L(G) = L(H)\}$$

?

- Para AFDs podemos usar o procedimento de decisão para V_{AFD} para provar que EQ_{AFD} é decidível.
- Para GLCs não é possível ... (Por quê?)
  ... Por que GLCs não são fechados sobre o complemento e sobre a intersecção.
  Mais tarde veremos que EQ_{GLC} não é decidível.

Decidibilidade

Teorema 4.9: Toda linguagem livre-de-contexto é decidível.

Ideia da prova:

- Seja A uma linguagem livre-de-contexto. Nosso objetivo é mostrar que A é decidível.
- Para isso, usaremos a MT S do Teorema 4.7 que decide A_{GLC}.

Decidibilidade

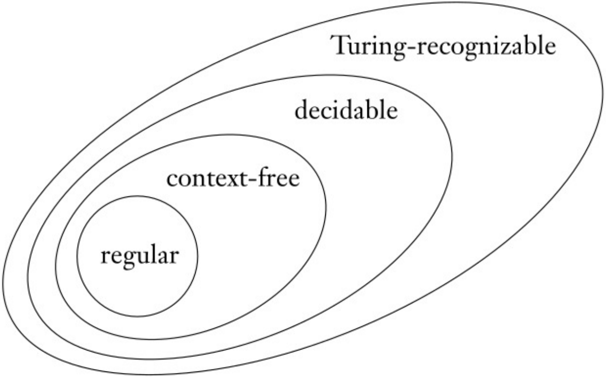
Teorema 4.9: Toda linguagem livre-de-contexto é decidível.

Prova: Utilizaremos a MT S que decide A_{GLC}. Seja G uma GLC para A e projetemos uma MT M_G que decide A.
$M_G =$ "Sobre a entrada w :

1. Rode a MT S sobre a entrada $\langle G, w \rangle$.
2. Se essa máquina aceita, *aceite*; se ela rejeita, *rejeite*."

Decidibilidade

O Teorema 4.9 provê a ligação final do relacionamento entre as quatro principais classes de linguagem.



Propriedade do Fechamento - União

Sejam L_1 e L_2 linguagens decidíveis, então a união

$$L = L_1 \cup L_2$$

é também decidível.

Prova: Mostramos decidibilidade de L construindo um decisor para ele. Seja M1 e M2 decisores para L1 e L2, respectivamente, então podemos construir um decisor M para L como segue:
M = "Sobre a entrada w,

1. Para cada caminho divida w em duas partes, $w = w_1 w_2$, faça
2. Execute M1 sobre w_1. Se ela aceita, aceite.
3. Execute M2 sobre w_2. Se ela aceita, aceite. Caso contrário, rejeite.

Propriedade do Fechamento - Concatenação

Sejam L_1 e L_2 linguagens decidíveis, então a concatenação

$$L = L_1 \cdot L_2$$

é também decidível.

Prova: Mostramos decidibilidade de L construindo um decisor para ele. Seja M1 e M2 decisores para L1 e L2, respectivamente, então podemos construir um MTN M para L como segue:

M = "Sobre a entrada w,

1. Para cada maneira de dividir w em duas partes, $w = w_1 w_2$, faça
2. Execute M1 sobre w_1
3. Execute M2 sobre w_2
4. Se qualquer combinação M1 e M2 aceita, aceite; caso contrário, rejeite."

Propriedade do Fechamento - Complemento

Seja L uma linguagem decidível, então o seu complemento

$$L' = \overline{L}$$

é também decidível.

Para qualquer linguagem decidível L, seja M uma MT que a decide. Construímos uma MT M' que decide L':
$M' =$ "Sobre a entrada w:

1. Execute M sobre w. Se M aceita, rejeite; Se M rejeita, *aceite*."

Como M' faz o oposto que qualquer M faz, então ele decide L'.

Sumário

Exercícios

1) Considere o problema de determinar se um AFD e uma expressão regular são equivalentes. Expresse este problema como linguagem e mostre que é decidível.

Exercícios (Solução)

Solução: Formulamos o problema como $EQ_{AFD,ER} = \{\langle A, R\rangle | A$ é um AFD, R é uma expressão regular e $L(A) = L(R)\}$. Construa a máquina T que decide $EQ_{AFD,ER}$.
T = "Sobre a entrada $\langle A, R\rangle$ onde A é AFD e R é expressão regular

1. Converta R em um AFD B equivalente. Portanto, $L(B) = L(R)$.
2. Execute a MT F que decide EQ_{AFD} sobre a entrada $\langle A, B\rangle$.
3. Aceite se F aceita, e rejeite caso contrário.

Exercícios

2) Seja $TODAS_{AFD} = \{\langle A \rangle | A$ é um AFD e $L(A) = \Sigma^*\}$ mostre que $TODAS_{AFD}$ é decidível.

Exercícios (Solução)

Solução: Construiremos uma MT M que decida $TODAS_{AFD}$. Para isso usaremos o decisor T que decide V_{AFD}
M = "Sobre a entrada $\langle A \rangle$ onde A é um AFD:

1. Construir um AFD B tal que $L(B) = \overline{L(A)}$.
2. Execute T sobre a entrada $\langle B \rangle$. Dê como saída o que T der como saída.

Como V_{AFD} é decidível, $TODAS_{AFD}$ é decidível.

Exercícios

3) Seja $INFINITA_{AFD} = \{\langle A\rangle | A$ é um AFD e $L(A)$ é uma linguagem infinita$\}$. Mostre que $INFINITA_{AFD}$ é decidível.

Exercícios (Solução)

Solução: A seguinte máquina de Turing M decide $INFINITA_{AFD}$.
M = "Sobre a entrada $\langle A \rangle$ onde A é AFD

1. Seja k o número de estados de A
2. Construa um AFD D que aceite todas as cadeias de comprimento k ou mais.
3. Construa um AFD M tal que $L(M) = L(A) \cap L(D)$
4. Teste $L(M) = \emptyset$, usando o decisor T de V_{AFD}
5. Se T aceita, rejeite; se T rejeita, aceite

Uma cadeia de comprimento k ou mais, onde k é o número de estados do AFD pode ser bombeada de maneira prescrita no lema de bombeamento para linguagens regulares para se obter uma quantidade infinita de cadeias aceitas.

Exercícios

4) Seja $A = \{\langle M \rangle | M$ é um AFD que não aceita nenhuma cadeia contendo um número ímpar de 1s$\}$. Mostre que A é decidível.

Exercícios (Solução)

A seguinte MT M decide A

M = "Sobre a entrada $\langle M \rangle$:

1. Construa um AFD O que aceite toda cadeia contendo um número ímpar de 1s.
2. Construa o AFD B tal que $L(B) = L(M) \cap L(O)$.
3. Teste se $L(B) = \emptyset$, usando o decisor T de V_{AFD}.
4. Se T aceita, aceite; se T rejeita, rejeite

Exercícios (Para casa)

5) Seja $A = \{\langle R, S\rangle |$ R e S são expressões regulares e $L(R) \subseteq L(S)\}$. Mostre que A é decidível.
6) Seja $INFINITA_{AP} = \{\langle M\rangle | M$ é um AP e $L(M)$ é uma linguagem infinita$\}$. Mostre que $INFINITA_{AP}$ é decidível.