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: 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: 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: 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 aceita, *aceite*, se N rejeita, *rejeite*"

Decidibilidade

Teorema: A linguagem

$$V_{AFD} = \{\langle A\rangle | A \text{ é um AFD e L(A)} = 0\}$$

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: 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, L(C) = $\emptyset$ implica L(A) = 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

Podemos construir um decisor M_{EQDFA} para EQ_{AFD}.
M_{EQDFA} = "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: 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 uma gramática equivalente na 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*"

Obs.: A computação é limitada por $2n-1 \rightarrow$ decisor.

Decidibilidade

Teorema: A linguagem

$$V_{GLC} = \{\langle G\rangle | \text{G é uma GLC e } L(G) = 0\}$$

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 $A \rightarrow U_1U_2\cdots U_k$ e cada símbolo $U_1U_2\cdots U_k$ 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: 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*."

Concatenação

Proposição : Sejam L1 e L2 linguagens decidíveis, então a concatenação L = L1 . L2 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
3. Execute M2 sobre w_2
4. Se qualquer combinação M1 e M2 aceita, aceite; caso contrário, rejeite."

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 um AFD e R é uma expressão regular

1. Converta R em um AFD B equivalente. Portanto, $L(B) = L(R)$.
2. Execute a MT $MEQDFA$ que decide EQ_{AFD} sobre a entrada $\langle A, B\rangle$.
3. Aceite se $MEQDFA$ 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 $L(M) = L(A) \cap L(B)$
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.