Disciplinas MCTA015-13 / CCM-104 Linguagens Formais e Autômata / Teoria da Computação

Profa. Carla Negri Lintzmayer

carla.negri@ufabc.edu.br

www.professor.ufabc.edu.br/∼carla.negri

Centro de Matemática, Computação e Cognição – Universidade Federal do ABC

UFABC

Aviso legal 👍

Estes slides não contêm um conteúdo completo sobre MTs: são apenas um apoio gráfico a uma aula específica dada em sala.

Decidibilidade

Mais problemas decidíveis

Problema: Dados um AFD e uma cadeia, ele a aceita?

Mais problemas decidíveis

Problema: Dados um AFD e uma cadeia, ele a aceita?

Linguagem: $A_{AFD} = \{\langle A, \omega \rangle \colon A$ é AFD que aceita $\omega\}$

Mais problemas decidíveis

Problema: Dados um AFD e uma cadeia, ele a aceita?

Linguagem: $A_{AFD} = \{\langle A, \omega \rangle \colon A$ é AFD que aceita $\omega\}$

$M_{A_{AFD}} =$ "`Sobre a entrada` $\langle A, \omega \rangle$`, onde` A `é AFD e` ω `é cadeia:`

1. `Simule` A `sobre` ω`.`
2. `Se a simulação terminar em um estado final, aceite. Caso contrário, rejeite.`"

Mais problemas decidíveis

Problema: Dados um AFD e uma cadeia, ele a aceita?

Linguagem: $A_{AFD} = \{\langle A, \omega\rangle \colon A \text{ é AFD que aceita } \omega\}$

$M_{A_{AFD}} =$ "`Sobre a entrada` $\langle A, \omega\rangle$`, onde` A `é AFD e` ω `é cadeia:`

1. **Simule** A `sobre` ω.
2. `Se a simulação terminar em um estado final, aceite. Caso contrário, rejeite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dados um AFN e uma cadeia, ele a aceita?

Mais problemas decidíveis

Problema: Dados um AFN e uma cadeia, ele a aceita?

Linguagem: $A_{AFN} = \{\langle A, \omega \rangle \colon A \text{ é AFN que aceita } \omega\}$

Mais problemas decidíveis

Problema: Dados um AFN e uma cadeia, ele a aceita?

Linguagem: $A_{AFN} = \{\langle A, \omega\rangle \colon A$ é AFN que aceita $\omega\}$

$M_{A_{AFN}} =$ "Sobre a entrada $\langle A, \omega\rangle$, onde A é AFN e ω é cadeia:

1. Converta A em um AFD B equivalente.
2. Execute $M_{A_{AFD}}$ sobre $\langle B, \omega\rangle$.
3. Se $M_{A_{AFD}}$ aceitar, aceite. Caso contrário, rejeite."

Mais problemas decidíveis

Problema: Dados um AFN e uma cadeia, ele a aceita?

Linguagem: $A_{AFN} = \{\langle A, \omega \rangle \colon A$ é AFN que aceita $\omega\}$

$M_{A_{AFN}} =$ "Sobre a entrada $\langle A, \omega \rangle$, onde A é AFN e ω é cadeia:

1. Converta A em um AFD B equivalente.
2. **Execute** $M_{A_{AFD}}$ sobre $\langle B, \omega \rangle$.
3. Se $M_{A_{AFD}}$ aceitar, aceite. Caso contrário, rejeite."

Por que é decisora?

Mais problemas decidíveis

Problema: Dados uma ER e uma cadeia, ela a descreve?

Mais problemas decidíveis

Problema: Dados uma ER e uma cadeia, ela a descreve?

Linguagem: $A_{ER} = \{\langle R, \omega\rangle\colon R$ é ER que descreve $\omega\}$

Mais problemas decidíveis

Problema: Dados uma ER e uma cadeia, ela a descreve?

Linguagem: $A_{ER} = \{\langle R, \omega \rangle \colon R$ é ER que descreve $\omega\}$

$M_{A_{ER}} =$ "`Sobre a entrada` $\langle R, \omega \rangle$`, onde` R `é ER e` ω `é cadeia:`

1. `Converta` R `em um AFN` A `equivalente.`
2. `Execute` $M_{A_{AFN}}$ `sobre` $\langle A, \omega \rangle$.
3. `Se` $M_{A_{AFN}}$ `aceitar, aceite. Caso contrário, rejeite.`"

Mais problemas decidíveis

Problema: Dados uma ER e uma cadeia, ela a descreve?

Linguagem: $A_{ER} = \{\langle R, \omega \rangle \colon R$ é ER que descreve $\omega\}$

$M_{A_{ER}} =$ "`Sobre a entrada` $\langle R, \omega \rangle$, `onde` R `é ER e` ω `é cadeia:`

1. `Converta` R `em um AFN` A `equivalente.`
2. `Execute` $M_{A_{AFN}}$ `sobre` $\langle A, \omega \rangle$.
3. `Se` $M_{A_{AFN}}$ `aceitar, aceite. Caso contrário, rejeite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dado um AFD, ele aceita alguma cadeia?

Mais problemas decidíveis

Problema: Dado um AFD, ele aceita alguma cadeia?

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

Mais problemas decidíveis

Problema: Dado um AFD, ele aceita alguma cadeia?

Linguagem: $V_{AFD} = \{\langle A \rangle \colon A$ é um AFD e $L(A) = \emptyset\}$

$M_{V_{AFD}} =$ "Sobre a entrada $\langle A \rangle$, onde A é AFD:

1. Marque o estado inicial de A.
2. Repita até que nenhum estado novo seja marcado:
   2.1 Marque qualquer estado que tenha uma transição para ele saindo de um marcado.
3. Se nenhum estado final estiver marcado, aceite. Caso contrário, rejeite."

Problema: Dado um AFD, ele aceita alguma cadeia?

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

$M_{V_{AFD}} =$ "`Sobre a entrada` $\langle A \rangle$, `onde` A `é AFD:`

1. `Marque o estado inicial de` A.
2. `Repita até que nenhum estado novo seja marcado:`
   2.1 `Marque qualquer estado que tenha uma transição para ele saindo de um marcado.`
3. `Se nenhum estado final estiver marcado, aceite. Caso contrário, rejeite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dado um AFD, ele aceita alguma cadeia?

Linguagem: $V'_{AFD} = \{\langle A \rangle : A$ é um AFD e $L(A) \neq \emptyset\}$

$M_{V'_{AFD}} =$ "Sobre a entrada $\langle A \rangle$, onde A é AFD:

1. Marque o estado inicial de A.
2. Repita até que nenhum estado novo seja marcado:
   2.1 Marque qualquer estado que tenha uma transição para ele saindo de um marcado.
3. Se nenhum estado final estiver marcado, rejeite. Caso contrário, aceite."

Mais problemas decidíveis

Problema: Dado um AFD, ele aceita alguma cadeia?

Linguagem: $V'_{AFD} = \{\langle A \rangle : A$ é um AFD e $L(A) \neq \emptyset\}$

$M_{V'_{AFD}} =$ "`Sobre a entrada` $\langle A \rangle$`, onde` A `é AFD:`

1. `Marque o estado inicial de` A.
2. `Repita até que nenhum estado novo seja marcado:`
   2.1 `Marque qualquer estado que tenha uma transição para ele saindo de um marcado.`
3. `Se nenhum estado final estiver marcado, rejeite. Caso contrário, aceite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dados dois AFDs, eles são equivalentes?

Mais problemas decidíveis

Problema: Dados dois AFDs, eles são equivalentes?

Linguagem: $E_{AFD} = \{\langle A, B\rangle \colon A$ e B são AFDs e $L(A) = L(B)\}$

Mais problemas decidíveis

Problema: Dados dois AFDs, eles são equivalentes?

Linguagem: $E_{AFD} = \{\langle A, B\rangle : A$ e B são AFDs e $L(A) = L(B)\}$

$M_{E_{AFD}} =$ "`Sobre a entrada` $\langle A, B\rangle$`, onde` A `e` B `são AFDs:`

1. `Construa um AFD` C `tal que` $L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$.
2. `Execute` $M_{V_{AFD}}$ `sobre` $\langle C\rangle$.
3. `Se` $M_{V_{AFN}}$ `aceitar, aceite. Caso contrário, rejeite.`"

Mais problemas decidíveis

Problema: Dados dois AFDs, eles são equivalentes?

Linguagem: $E_{AFD} = \{\langle A, B\rangle \colon A$ e B são AFDs e $L(A) = L(B)\}$

$M_{E_{AFD}} =$ "`Sobre a entrada` $\langle A, B\rangle$, `onde` A `e` B `são AFDs:`

1. `Construa um AFD` C `tal que` $L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$.
2. `Execute` $M_{V_{AFD}}$ `sobre` $\langle C\rangle$.
3. `Se` $M_{V_{AFN}}$ `aceitar, aceite. Caso contrário, rejeite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle \colon G$ é GLC e ω é cadeia$\}$

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle \colon G$ é GLC e ω é cadeia$\}$

$T =$ "`Sobre a entrada` $\langle G, \omega \rangle$`, onde` G `é GLC e` ω `é cadeia:`

1. `Para cada derivação de` G`:`
   1.1 `Se gerou` ω`, aceite.`
2. `Rejeite.`"

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle \colon G$ é GLC e ω é cadeia$\}$

$T =$ "`Sobre a entrada` $\langle G, \omega \rangle$`, onde` G `é GLC e` ω `é cadeia:`

1. `Para cada derivação de` G:
    1.1 `Se gerou` ω`, aceite.`
2. `Rejeite.`"

Por que é decisora?

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle\colon G$ é GLC e ω é cadeia$\}$

$T =$ "`Sobre a entrada` $\langle G, \omega \rangle$`, onde` G `é GLC e` ω `é cadeia:`

1. `Para cada derivação de` G`:`
   1.1 `Se gerou` ω`, aceite.`
2. `Rejeite.`"

Por que é decisora? Não é!!

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle \colon G$ é GLC e ω é cadeia$\}$

Mais problemas decidíveis

Problema: Dadas uma GLC e uma cadeia, ela a gera?

Linguagem: $A_{GLC} = \{\langle G, \omega \rangle \colon G$ é GLC e ω é cadeia$\}$

$M_{A_{GLC}} =$ "Sobre a entrada $\langle G, \omega \rangle$, onde G é GLC e ω é cadeia:

1. Converta G em uma GLC G' equivalente na FNC.
2. Se $|\omega| = 0$, faça $m = 1$. Caso contrário, faça $m = 2|\omega| - 1$.
3. Para cada derivação de G com até m passos:
   3.1 Se gerou ω, aceite.
4. Rejeite."

Mais problemas decidíveis

Problema: Dada uma GLC, ela gera alguma cadeia?

Mais problemas decidíveis

Problema: Dada uma GLC, ela gera alguma cadeia?

Linguagem: $V_{GLC} = \{\langle G \rangle \colon G$ é GLC e $L(G) = \emptyset\}$

Mais problemas decidíveis

Problema: Dada uma GLC, ela gera alguma cadeia?

Linguagem: $V_{GLC} = \{\langle G \rangle : G$ é GLC e $L(G) = \emptyset\}$

$M_{V_{GLC}} =$ "`Sobre a entrada` $\langle G \rangle$, `onde` G `é GLC:`

1. `Marque todos os terminais.`
2. `Repita até que nenhuma variável seja marcada:`
   2.1 `Marque uma variável` A `tal que existe uma regra` $A \to \alpha$ `em que todo símbolo de` α `esteja marcado.`
3. `Se a variável inicial está marcada, rejeite. Caso contrário, aceite.`"

LLCs e MTs

Relação final entre as linguagens

Teorema

Toda linguagem livre de contexto é Turing-Decidível.

Relação final entre as linguagens

Teorema

Toda linguagem livre de contexto é Turing-Decidível.

Demonstração: Seja A uma LLC. Então existe uma GLC G que a gera.

A MT a seguir é decisora para A:

Relação final entre as linguagens

Teorema

Toda linguagem livre de contexto é Turing-Decidível.

Demonstração: Seja A uma LLC. Então existe uma GLC G que a gera.

A MT a seguir é decisora para A:

$M_A =$ "`Sobre a entrada` ω`:`

1. `Execute` $M_{A_{GLC}}$ `sobre` $\langle G, \omega \rangle$`.`
2. `Se` $M_{A_{GLC}}$ `aceitar, aceite. Caso contrário, rejeite.`"