Teoria da Computação

Prof. Maicon R. Zatelli

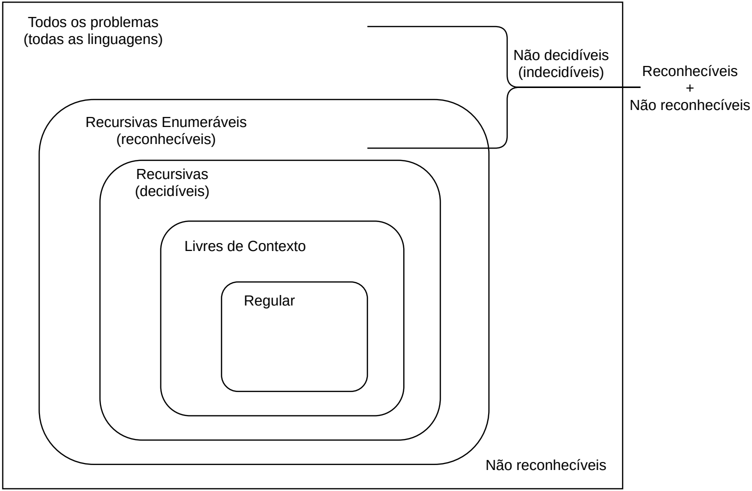
Aula 8 - Revisão P2

Universidade Federal de Santa Catarina
Florianópolis - Brasil

Introdução

- Decidibilidade (Linguagens Decidíveis/Linguagens Indecidíveis)
- Redutibilidade

Hierarquia de Chomsky



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

Linguagens Decidíveis

1. $L = f < D, N > j\ D$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \qquad L(N)\ g$

2. $L = f < G, w > j\ G$ é uma Gramática Regular e G gera a palavra $w\ g$

3. $L = f < R > j\ R$ é uma Expressão Regular e $L(R) = ;\ g$

4. $L = f < R, D > j\ R$ é uma Expressão Regular e D é um Autômato Finito Determinístico e $L(R) = L(D)\ g$

5. $L = f < A, B > j\ A$ e B são Autômatos Finitos Não-Determinísticos e $L(A)\ [\ L(B) = ;\ g$

Linguagens Decidíveis

Linguagem: $L = f< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Linguagens Decidíveis

Linguagem: $L = f\!< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Linguagens Decidíveis

Linguagem: $L = f\!< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

Linguagens Decidíveis

Linguagem: $L = f< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

- Note que $L(D) \quad L(N)$ significa que $L(D) \setminus \overline{L(N)} = ;$

Linguagens Decidíveis

Linguagem: $L = \{\langle D, N \rangle \mid D$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \subseteq L(N)\}$

Teorema: L é decidível

Prova: por construção

- Note que $L(D) \subseteq L(N)$ significa que $L(D) \cap \overline{L(N)} = \emptyset$
- Ou seja, não há nenhum elemento que esteja em $L(D)$, mas não esteja em $L(N)$

Linguagem: $L = f< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

- Note que $L(D) \quad L(N)$ significa que $L(D) \backslash \overline{L(N)} = ;$
- Ou seja, não há nenhum elemento que esteja em $L(D)$, mas não esteja em $L(N)$
- Assim, podemos construir um AFD A que aceite a linguagem $L(D) \backslash \overline{L(N)}$

Linguagens Decidíveis

Linguagem: $L = f\!< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

- Note que $L(D) \quad L(N)$ significa que $L(D) \setminus \overline{L(N)} = ;$
- Ou seja, não há nenhum elemento que esteja em $L(D)$, mas não esteja em $L(N)$
- Assim, podemos construir um AFD A que aceite a linguagem $L(D) \setminus \overline{L(N)}$
- Isso é possível pois as linguagens regulares são fechadas nas operações de intersecção e complemento

Linguagens Decidíveis

Linguagem: $L = f\!< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

- Note que $L(D) \quad L(N)$ significa que $L(D) \setminus \overline{L(N)} = ;$
- Ou seja, não há nenhum elemento que esteja em $L(D)$, mas não esteja em $L(N)$
- Assim, podemos construir um AFD A que aceite a linguagem $L(D) \setminus \overline{L(N)}$
- Isso é possível pois as linguagens regulares são fechadas nas operações de intersecção e complemento
- Por fim, construímos uma MT que decide se $L(A) = ;$ (ou usamos o decisor de E_{DFA})

Linguagens Decidíveis

Linguagem: $L = f\!< D, N > jD$ é um Autômato Finito Determinístico e N é um Autômato Finito Não-Determinístico e $L(D) \quad L(N)g$

Teorema: L é decidível

Prova: por construção

Seja M uma MT que decide L e ME uma MT que decide E_{DFA}

```
M com a entrada <D,N>, onde D é um AFD e N é um AFND, faz:
    1: Construa um AFD A que reconheça a linguagem L(D) \ L(N)
    2: Rode ME (decisor de E_DFA) com a entrada <A>
    3: Se ME aceita <A>, aceite (M aceita <D,N>), caso contrário,
       rejeite (M rejeita <D,N>)
```

Linguagens Decidíveis

- $L = f < G, w > j\ G$ é uma Gramática Regular e G gera a palavra $w\ g$

Linguagens Decidíveis

- $L = f < G, w > j$ G é uma Gramática Regular e G gera a palavra w g
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}

Linguagens Decidíveis

- $L = f < G, w > j\ G$ é uma Gramática Regular e G gera a palavra $w\ g$
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = f < R > j\ R$ é uma Expressão Regular e $L(R) = ;\ g$

Linguagens Decidíveis

- $L = f < G, w > j$ G é uma Gramática Regular e G gera a palavra w g
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = f < R > j$ R é uma Expressão Regular e $L(R) = ;$ g
  - Converta a ER em um AFD e depois é a mesma prova de E_{DFA}

Linguagens Decidíveis

- $L = f < G, w > j\ G$ é uma Gramática Regular e G gera a palavra $w\ g$
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = f < R > j\ R$ é uma Expressão Regular e $L(R) = ;\ g$
  - Converta a ER em um AFD e depois é a mesma prova de E_{DFA}
- $L = f < R, D > j\ R$ é uma Expressão Regular e D é um Autômato Finito Determinístico e $L(R) = L(D)\ g$

Linguagens Decidíveis

- $L = f < G, w > j$ G é uma Gramática Regular e G gera a palavra w g
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = f < R > j$ R é uma Expressão Regular e $L(R) = ;$ g
  - Converta a ER em um AFD e depois é a mesma prova de E_{DFA}
- $L = f < R, D > j$ R é uma Expressão Regular e D é um Autômato Finito Determinístico e $L(R) = L(D)$ g
  - Converta a ER em um AFD e depois é a mesma prova de EQ_{DFA}

Linguagens Decidíveis

- $L = \{ < G, w > \mid G$ é uma Gramática Regular e G gera a palavra $w \}$
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = \{ < R > \mid R$ é uma Expressão Regular e $L(R) = \emptyset \}$
  - Converta a ER em um AFD e depois é a mesma prova de E_{DFA}
- $L = \{ < R, D > \mid R$ é uma Expressão Regular e D é um Autômato Finito Determinístico e $L(R) = L(D) \}$
  - Converta a ER em um AFD e depois é a mesma prova de EQ_{DFA}
- $L = \{ < A, B > \mid A$ e B são Autômatos Finitos Não-Determinísticos e $L(A) \cup L(B) = \emptyset \}$

Linguagens Decidíveis

- $L = f < G, w > j\ G$ é uma Gramática Regular e G gera a palavra $w\ g$
  - Converta a GR em um AFD e depois é a mesma prova de A_{DFA}
- $L = f < R > j\ R$ é uma Expressão Regular e $L(R) = ;\ g$
  - Converta a ER em um AFD e depois é a mesma prova de E_{DFA}
- $L = f < R, D > j\ R$ é uma Expressão Regular e D é um Autômato Finito Determinístico e $L(R) = L(D)\ g$
  - Converta a ER em um AFD e depois é a mesma prova de EQ_{DFA}
- $L = f < A, B > j\ A$ e B são Autômatos Finitos Não-Determinísticos e $L(A)\ [\ L(B) = ;\ g$
  - Crie um AFD que reconheça a linguagem $L(A)\ [\ L(B)$ e depois é a mesma prova de E_{DFA}

Linguagens Indecidíveis

1. $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$

2. $NE_{TM} = f < M > j\ M$ é uma Máquina de Turing e $L(M)$ 6= ; g

3. $Rot_{TM} = f < M > j\ M$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $rotaciona\_1(w)\ 2\ L(M)\ g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita
   - *Exemplo*: a rotação circular da palavra *abcde* em uma posição para a direita resulta em *eabcd*. Caso uma nova rotação seja feita, então a palavra resultante é *deabc*.

4. $Rev_{MT} = f < M > j\ M$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $w^R\ 2\ L(M)\ g$

Redutibilidade

Linguagem: $Rot_{TM} = f < M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Redutibilidade

Linguagem: $Rot_{TM} = f < M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Redutibilidade

Linguagem: $Rot_{TM} = f< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

Redutibilidade

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $rotaciona\_1(w)\ 2\ L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos reduzir A_{TM} para Rot_{TM}, sabendo que A_{TM} é indecidível

Redutibilidade

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $rotaciona\_1(w)\ 2\ L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos reduzir A_{TM} para Rot_{TM}, sabendo que A_{TM} é indecidível
- Suponha que Rot_{TM} é decidível, então existe um decisor R para Rot_{TM}

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos reduzir A_{TM} para Rot_{TM}, sabendo que A_{TM} é indecidível
- Suponha que Rot_{TM} é decidível, então existe um decisor R para Rot_{TM}
- Podemos construir um decisor S para A_{TM} utilizando R como subrotina

Redutibilidade

Linguagem: $Rot_{TM} = f < M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos reduzir A_{TM} para Rot_{TM}, sabendo que A_{TM} é indecidível
- Suponha que Rot_{TM} é decidível, então existe um decisor R para Rot_{TM}
- Podemos construir um decisor S para A_{TM} utilizando R como subrotina
- A ideia é quando uma MT aceita uma linguagem de rotações significa que M aceita w

Redutibilidade

Linguagem: $Rot_{TM} = f< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos reduzir A_{TM} para Rot_{TM}, sabendo que A_{TM} é indecidível
- Suponha que Rot_{TM} é decidível, então existe um decisor R para Rot_{TM}
- Podemos construir um decisor S para A_{TM} utilizando R como subrotina
- A ideia é quando uma MT aceita uma linguagem de rotações significa que M aceita w
- Quando uma MT aceita uma linguagem faltando rotações significa que M não aceita w

Redutibilidade

Linguagem: $Rot_{TM} = \{< M > | M$ é uma Máquina de Turing e $w \in L(M)$ se e somente se $rotaciona\_1(w) \in L(M)\}$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $\{abc, cab, bca\}$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

Redutibilidade

Linguagem: $Rot_{TM} = f< M > jM$ é uma Máquina de Turing e w *2* $L(M)$ se e somente se $rotaciona\_1(w)$ *2* $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

- Vamos inicialmente precisar criar uma MT Mw que aceita uma linguagem de rotação ou uma linguagem faltando rotações

Redutibilidade

Linguagem: $Rot_{TM} = \{< M > | M$ é uma Máquina de Turing e $w \in L(M)$ se e somente se $rotaciona\_1(w) \in L(M)\}$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $\{abc, cab, bca\}$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

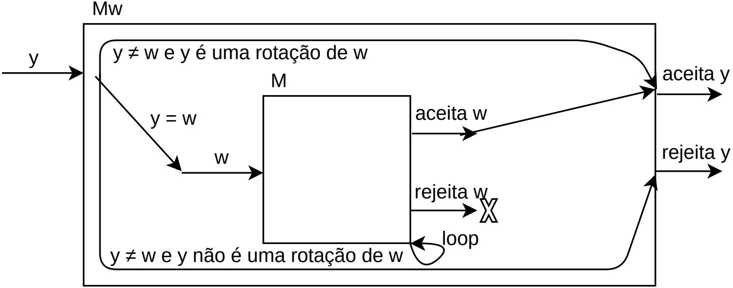
- Vamos inicialmente precisar criar uma MT Mw que aceita uma linguagem de rotação ou uma linguagem faltando rotações

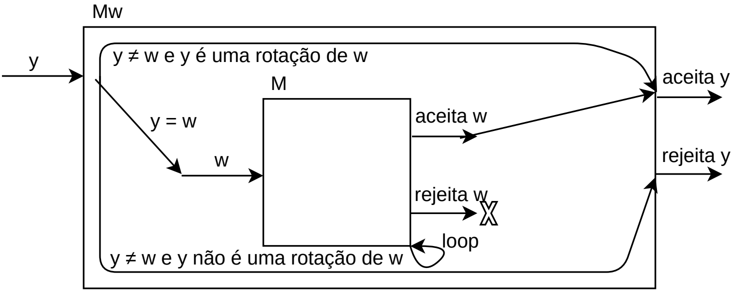
```
Mw com a entrada y, onde y é uma palavra, faz:
  1: Se y ≠ w e y = w rotacionado, aceite y
  2: Senão, se y = w, rode M com a entrada w
    2.1: Se M aceita w, aceite y
  3: Senão, se y ≠ w, rejeite y
```

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução



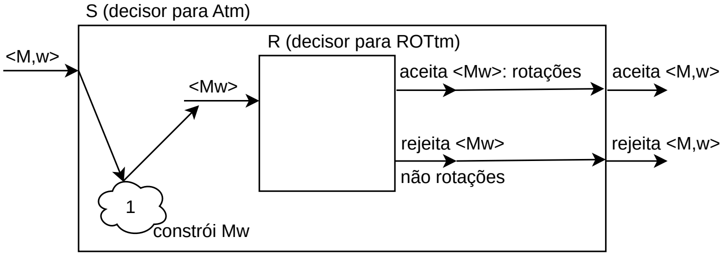


- *Mw* é uma MT que reconhece ou a linguagem com as rotações de *w* (inclusive *w*) ou a linguagem com as rotações de *w* (sem o *w*)
- Assim,
  - *Mw* aceita as rotações + *w* se e somente se *M* aceita *w*
  - *Mw* aceita as rotações - *w* se e somente se *M* não aceita *w*

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução



Redutibilidade

Linguagem: $Rot_{TM} = f < M > jM$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $rotaciona\_1(w)\ 2\ L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

```
S com a entrada <M,w>, onde M é uma MT e w é uma palavra, faz:
    1: Construa a MT Mw
    2: Rode R com a entrada <Mw>
        2.1: Se R aceita <Mw>, aceite (S aceita <M,w>)
        2.2: Se R rejeita <Mw>, rejeite (S rejeita <M,w>)
```

Redutibilidade

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

```
S com a entrada <M,w>, onde M é uma MT e w é uma palavra, faz:
    1: Construa a MT Mw
    2: Rode R com a entrada <Mw>
        2.1: Se R aceita <Mw>, aceite (S aceita <M,w>)
        2.2: Se R rejeita <Mw>, rejeite (S rejeita <M,w>)
```

- Portanto, se R decide Rot_{TM}, então S decide A_{TM}

Redutibilidade

Linguagem: $Rot_{TM} = f < M > jM$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $rotaciona\_1(w)\ 2\ L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

```
S com a entrada <M,w>, onde M é uma MT e w é uma palavra, faz:
    1: Construa a MT Mw
    2: Rode R com a entrada <Mw>
        2.1: Se R aceita <Mw>, aceite (S aceita <M,w>)
        2.2: Se R rejeita <Mw>, rejeite (S rejeita <M,w>)
```

- Portanto, se R decide Rot_{TM}, então S decide A_{TM}
- Sabemos que A_{TM} é indecidível e então há uma contradição

Redutibilidade

Linguagem: $Rot_{TM} = f\!< M > jM$ é uma Máquina de Turing e w 2 $L(M)$ se e somente se $rotaciona\_1(w)$ 2 $L(M)g$ onde, $rotaciona\_1(w)$ retorna a rotação circular de w em uma posição para a direita. Ex: $fabc, cab, bcag$

Teorema: Rot_{TM} é indecidível

Prova: por contradição, usando redução

```
S com a entrada <M,w>, onde M é uma MT e w é uma palavra, faz:
    1: Construa a MT Mw
    2: Rode R com a entrada <Mw>
        2.1: Se R aceita <Mw>, aceite (S aceita <M,w>)
        2.2: Se R rejeita <Mw>, rejeite (S rejeita <M,w>)
```

- Portanto, se R decide Rot_{TM}, então S decide A_{TM}
- Sabemos que A_{TM} é indecidível e então há uma contradição
- S não pode existir e assim R também não pode existir

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$
  - Como $;$ é uma linguagem regular, note que é fácil reduzir E_{TM} para X_{TM}
  - Basta verificar se M aceita a linguagem regular $;$

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$
  - Como $;$ é uma linguagem regular, note que é fácil reduzir E_{TM} para X_{TM}
  - Basta verificar se M aceita a linguagem regular $;$

- $NE_{TM} = f < M > j\ M$ é uma Máquina de Turing e $L(M) \neq\ ;\ g$

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$
  - Como $;$ é uma linguagem regular, note que é fácil reduzir E_{TM} para X_{TM}
  - Basta verificar se M aceita a linguagem regular $;$

- $NE_{TM} = f < M > j\ M$ é uma Máquina de Turing e $L(M)$ 6= $;\ g$
  - Novamente, note que podemos reduzir E_{TM} para NE_{TM}

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$
  - Como $;$ é uma linguagem regular, note que é fácil reduzir E_{TM} para X_{TM}
  - Basta verificar se M aceita a linguagem regular $;$

- $NE_{TM} = f < M > j\ M$ é uma Máquina de Turing e $L(M) \not= ;\ g$
  - Novamente, note que podemos reduzir E_{TM} para NE_{TM}

- $Rev_{MT} = f < M > j\ M$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $w^R\ 2\ L(M)\ g$

Linguagens Indecidíveis

- $X_{TM} = f < M, X > j\ M$ é uma Máquina de Turing e M aceita a linguagem regular $X\ g$
  - Como $;$ é uma linguagem regular, note que é fácil reduzir E_{TM} para X_{TM}
  - Basta verificar se M aceita a linguagem regular $;$

- $NE_{TM} = f < M > j\ M$ é uma Máquina de Turing e $L(M)\ 6\!\!= \ ;\ g$
  - Novamente, note que podemos reduzir E_{TM} para NE_{TM}

- $Rev_{MT} = f < M > j\ M$ é uma Máquina de Turing e $w\ 2\ L(M)$ se e somente se $w^R\ 2\ L(M)\ g$
  - Idem Rot_{TM}, mas ao invés de rotações de w temos o reverso de w

Conclusão

- Decidibilidade (Linguagens Decidíveis/Linguagens Indecidíveis)
- Redutibilidade