MATA51: Teoria da Computação

Semestre 2021.1

Prof. Laís Nascimento

Alberto Lucas e Renata Ribeiro

Lista de Exercícios 6 - Indecidibilidade & Redução: Linguagens RE (Recursivamente Enumerável), não recursivas e Linguagens não RE

---

1. **Utilizar reduções para mostrar que as seguintes linguagens são indecidíveis:**

a. $\overline{L_d} = \{\omega_i | \omega_i \in L(M_i)\}$

A linguagem é formada pelo conjunto de cadeias wi tal que wi não está em L(Mi), onde L(Mi) é o conjunto de cadeias aceitas pela máquina de Turing Mi.

Prova por contradição: Temos que $\overline{L_d}$ é decidível.

- Sabemos que a máquina de Turing Mi decide L(Mi).
- Construímos uma máquina de Turing R que usa Mi para decidir $\overline{L_d}$.

Para R com a entrada <Mi,wi>, temos:

- Chamamos R com entrada <Mi, wi>
- Se Mi aceita, R rejeita.
- Se Mi rejeita, R aceita.

Mas se Mi aceita uma palavra, essa mesma palavra não é aceita por R. Se R rejeita é porque não é Turing-reconhecível, o que é uma contradição, já que toda linguagem Turing-reconhecível é decidível. Portanto, $\overline{L_d}$ é indecidível como queríamos provar.

b. L_{MTdec} = {<M> | M é uma MT e L(M) é decidível}

Prova por contradição: Temos que L_{MTdec} é decidível.

- Suponhamos então que a máquina de Turing R decide L_{dec}.
- Construímos uma máquina de Turing S que usa R para decidir L_{MTdec}

Para S com a entrada <M,w>, temos:

- Chamamos R com entrada <M, w>
- Se R aceita, S aceita.
- Se R aceita, simule M sobre w até que ela pare
- Se R rejeita, S rejeita.

Mas se R rejeita é porque não é Turing-reconhecível, o que é uma contradição, já que toda linguagem Turing-reconhecível é decidível. Portanto S pode ser reduzida para R e como R é indecidível, S é indecidível.

c. L_{MTfin} = {<M> | M é uma MT e L(M) é uma linguagem infinita}.

Suponha L_{MTfin} seja decidida por *I* . Então podemos fazer um decisor D para a A_{MT}, do seguinte modo:

D = "Na entrada <M,w> onde M é uma MT:

- Construa uma MT M' = "Na entrada x:
  - i.Se x = ε, aceite.
  - ii.Se x ≠ ε, ececute M em w e aceite x somente se M aceita w."
- Execute *I* em <M'>.
- Se *I* aceitar, aceite; por outro lado, rejeite."

Se M aceita w, então L(M') = Σ*, na qual é uma linguagem infinita; se M não aceita w, então L(M') = {ε}, na qual é uma linguagem finita. Portanto, uma vez que A_{MT} é indecidível, então L_{MTfin} também é. Na verdade, isso mostra algo mais importante: dado uma linguagem $L \subseteq \Sigma^*$ checar se uma MT M arbitrária possui L(M) = L é indecidível.

2. Mostre que:

a. $\overline{L_d} = \{\omega_i | \omega_i \in L(M_i)\}$ é RE.

Se existe uma palavra wi tal que pertence a linguagem L(Mi) e essa palavra ao ser inserida em uma máquina de Turing é reconhecida, então a linguagem $\overline{L_d}$ é Turing-reconhecível, visto que se existe máquina de Turing que reconheça L(Mi) é possível também reconhecer as palavras que não fazem parte de L(Mi). Se $\overline{L_d}$ é Turing-reconhecível, $\overline{L_d}$ é RE.

b. $\overline{L_u} = \{\langle M \rangle \omega | M \text{ é uma MT que não aceita } \omega\}$ é não RE.

Se existe uma palavra w e essa palavra ao ser inserida em uma máquina de Turing não é aceita, então a linguagem $\overline{L_u}$ que é o complemento não é Turing-reconhecível e, portanto, não é RE.

c. $\overline{PARA_{MT}} = \{\langle M', \omega \rangle | M \text{ entra em loop com a entrada } \omega\}$ é não RE

Se a palavra w ao ser inserida na máquina de Turing M entra em loop, ela não é Turing-reconhecível e portanto não é RE.

d. D={p| p é um polinômio com raiz inteira} é RE

Tome M1 como uma entrada: um polinômio p sobre x. E siga os seguintes passos:

- Ao calcular o valor de p com x substituída sucessivamente por 0, 1, -1, 2, -2, 3, -3,... Se em algum ponto p = 0, aceite.
- M1 não terminará, se não houver raiz.

Para M1 ser um decisor, temos que calcular limitantes para as raízes de um polinômio de uma variável e restringir a busca a esses limitantes:

$$\pm \frac{k^{*}cmax}{c1}$$

onde **k = número de termos no polinômio**, **cmax = o coeficiente com o maior valor absoluto** e **c1 = o coeficiente do termo de mais alta ordem**.

Se uma raiz não for encontrada dentro desses limitantes, a máquina rejeita. Yuri Matijasevich mostrou em 1970 que calcular limitantes para polinômios multivariados é impossível.

Portanto, é Turing-reconhecível e portanto é RE.

3. **Explique o Teorema de Rice. Como ele pode ser usado em demonstrações de indecidibilidade?**

O teorema de Rice diz que todo o teste de qualquer propriedade que não sejam triviais relativas a linguagens reconhecidas por máquinas de Turing é indecidível. Uma propriedade é dita trivial se ela é vazia ou é correspondente a um conjunto de linguagens RE. Exemplo:

Seja P um problema sobre máquinas de Turing satisfazendo as seguintes propriedades:

- para quaisquer máquinas M1 e M2, onde L(M1) = L(M2), temos que (M1) pertence a P se e somente se (M2) pertence a P. Em outras palavras, M pertence a P dependendo apenas da linguagem reconhecida por M;

- Existem M1 e M2 tais que (M1) pertence a P e (M2) não pertence a P. Em outras palavras, P não é trivial.

Referências

1. PASSOS, Yure, Teorema de Rice, Slides Share, 2018. Disponível em: <https://pt.slideshare.net/yuripassos58/teorema-derice-108851397>. Acesso em 07 de Agosto de 2018.

2. Decidibilidade. UFPE, 2018. Disponível em <encurtador.com.br/ouvCE>. Acesso em: 20 de Junho de 2018.

3. Decidibilidade. UFPE, 2018. Disponível em <encurtador.com.br/pELNS>. Acesso em: 20 de Junho de 2018.

4. Complexidade Computacional. USP, 2010. Disponível em <https://www.ime.usp.br/~coelho/mac5722-2010/aulas/aula04.pdf>. Acesso em: 17 de Março de 2010.

5. Linguagem recursivamente enumerável. Wikipédia, 2017. Disponível em <https://pt.wikipedia.org/wiki/Linguagem_recursivamente_enumer%C3%A1vel>. Acesso em: 8 de Outubro de 2017.

6. LONGO, Humberto, Redutibilidade, INF/UFG, 2012. Disponível em: <https://ava.ufba.br/pluginfile.php/1142023/mod_folder/content/0/%282%29Reduc%CC%A7a%CC%83o.pdf?forcedownload=1>. Acesso em 12 de Fevereiro de 2012.

7. LIMA, Ariane Machado, Problemas Indecidíveis, USP, 2020. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4480390/mod_resource/content/1/ACH2043-Aula23-Cap5.1e5.3-ProblemasIndecid%C3%ADveis_e_Redutibilidade_por_mapeamento.pdf>. Acesso em 24 de Novembro de 2020.

8. COLBOURN, Advanced Theory of Computation. Public.asu.edu, 2016. Disponível em <http://www.public.asu.edu/~ccolbou/src/355quiz14f16sol.pdf>. Acesso em: 5 de Março de 2016.