Indecidibilidade

Prof^a Jerusa Marchi

Departamento de Informática e Estatística

Universidade Federal de Santa Catarina

e-mail: jerusa@inf.ufsc.br

Indecidibilidade

- Antes de vermos algumas provas de indecidibilidade, precisamos retomar alguns conceitos importantes:
  - 1^o) O conjunto das possíveis máquinas de Turing é enumerável
    - Qualquer máquina de Turing M pode ser representada como uma "codificação", ou seja $\langle M \rangle$
    - O conjunto de todas as cadeias Σ^* é enumerável para qualquer alfabeto Σ
    - Todas as codificações legítimas de MT formam um subconjunto próprio de Σ^*, portanto, enumerável.

Indecidibilidade

- Antes de vermos algumas provas de indecidibilidade, precisamos retomar alguns conceitos importantes:
  - 2^o) O conjunto de todas as linguagens é não enumerável
    - dado o conjunto L de todas as possíveis linguagens ($L = 2^{\Sigma^*}$)
    - dado o conjunto não enumerável de todas as sequências binárias infinitas $\mathcal{B}$
    - cada linguagem em L é representada por uma sequência característica infinita $\mathcal{X}_A$, tal que $\mathcal{X}_A \in \mathcal{B}$
    - portanto, há uma $f : \mathcal{L} \mapsto \mathcal{B}$ que é bijetora, sendo $\mathcal{B}$ incontável, $\mathcal{L}$ também é incontável.
- Com isso demonstramos que há algumas linguagens que não são Turing-Reconhecíveis, e portanto há problemas que não são computáveis.

Indecidibilidade

- Pela Tese de Church-Turing, tudo o que é **computável** é computável por uma máquina de Turing
- Dentre os problemas computáveis (ou seja, para os quais há uma MT), quais são "efetivamente" computáveis (ou seja, decidíveis)?
  - O que queremos de fato averiguar é se o conjunto das Linguagens Recursivas é um subconjunto próprio das Linguagens Recursivamente Enumeráveis.

Indecidibilidade

- Problema da Parada: dada uma máquina M e qualquer entrada w, M pára ao computar w?
  - Quando Turing desenvolveu a prova, a MT aceitava por parada e não por estado final.
  - Turing desenvolveu uma prova por contradição, usando o argumento da diagonalização

O Problema da Parada

- Inicialmente concebe-se uma MT U que é um decisor (sempre pára voltando sim ou não)
- Sobre U são feitas alterações simples (todas “realizáveis” em qualquer linguagem de programação)
  - A MT U' recebe $\langle M, w\rangle$ e entra em loop se M pára ao computar w e pára se M entra em loop
  - A MT U'' recebe somente $\langle M\rangle$, duplica $\langle M\rangle$ e chama U’
- A grande “sacada” de Turing ao construir a prova, foi oferecer à Máquina uma cópia de si mesma como entrada (como um compilador de C que foi ele próprio escrito em C).
  - Se U'' pára é por que U'' entrou em loop
  - Se U'' entra em loop é por que U'' pára (contradição).

Indecidibilidade

- CONCLUSÃO:
  - A existência de uma MT Universal (U), ou seja, uma máquina de Turing que simula uma MT M para uma entrada w, prova que $L(U)$, ou seja a linguagem formada por todas as Máquinas de Turing e todas as entradas, é recursivamente enumerável
  - Ao mostrar a indecidibilidade do problema da parada, mostra-se que não existe uma MT que sempre páre e que seja equivalente a U, ou seja, que $L(U)$ não é recursiva
  - A classe das linguagens recursivas é uma subconjunto próprio da classe das linguagens recursivamente enumeráveis

Uma Linguagem Turing-Irreconhecível

- Após termos visto um problema que é indecidível, podemos exemplificar um problema que não é sequer Turing-Reconhecível
- para isto, vamos mostrar que se uma linguagem e o seu complemento forem ambos Turing-Reconhecíveis, então a linguagem é **decidível**
- Se uma linguagem é **indecidível**, ou ela ou o seu complemento é não Turing-Reconhecível (**não computável**)
  - Uma linguagem é dita **co-Turing-Reconhecível** se ela for o complemento de uma linguagem **Turing-Reconhecível**

Uma Linguagem Turing-Irreconhecível

- **Teorema:** Uma linguagem é decidível sse ela é Turing-Reconhecível e co-Turing-Reconhecível.
- **Prova:** ($\rightarrow$) Se A for decidível, é fácil ver que tanto A quanto $\overline{A}$ são Turing-reconhecíveis. Qualquer linguagem decidível é Turing-Reconhecível, e o complemento de uma linguagem decidível também é decidível.
  ($\leftarrow$) Se tanto A quanto $\overline{A}$ são Turing-Reconhecíveis. Fazemos M_1 ser o reconhecedor para A e M_2 o reconhecedor para $\overline{A}$. A MT M recebe a entrada w e
  1. Roda ambas M_1 e M_2 sobre a entrada w em paralelo
  2. Se M_1 aceita, M aceita; se M_2 aceita, M rejeita

  Uma vez que M pára sempre que M_1 ou M_2 aceita, M sempre pára e, portanto é um decisor. Além disso M aceita exatamente as palavras w pertencentes a A. Logo M é um decisor para A e consquentemente A é decidível.

Uma Linguagem Turing-Irreconhecível

- Seja $A_{MT} = \{\langle M, w\rangle \mid M$ é uma MT e M pára ao computar $w\}$
  - **Corolário:** A linguagem $\overline{A_{MT}}$ é não Turing-Reconhecível
  - **Prova:** Sabemos que A_{MT} é Turing-Reconhecível. Se $\overline{A_{MT}}$ também fosse Turing-Reconhecível, A_{MT} seria decidível. A prova do teorema da parada nos mostra que A_{MT} não é decidível, portanto $\overline{A_{MT}}$ não pode ser Turing-Reconhecível.