Indecidibilidade

Prof^a Jerusa Marchi

Departamento de Informática e Estatística

Universidade Federal de Santa Catarina

e-mail: jerusa@inf.ufsc.br

Computabilidade

- Tudo que pode ser representado de modo finito
- Tudo o que pode ser descrito através de um “procedimento algorítmico”
- Toda linguagem que é recursivamente enumerável (ou seja pode ser reconhecida por uma máquina de Turing - Enumerador)
- Toda função recursiva parcial (ou seja, que pode ser gerada a partir das funções básicas e fechada sob composição, recursão primitiva e iteração ilimitada)

Computabilidade

Linguagem	Gramática	Funções Numéricas	Máquinas
Recursivamente Enumerável	Irrestrita	F.R.Parcial	MT
Recursiva	Irrestrita	F.R.Totais	MT
Sensível ao Contexto	Sensível ao Contexto	F.R.Primitivas	MT

Computabilidade

- Tudo o que é **computável** é computável por uma máquina de Turing
  - Máquinas de Turing que param aceitando ou não a entrada
  - Máquinas de Turing que podem rodar para sempre com entradas que não são aceitas

Decidibilidade

- *Tomemos a Máquina de Turing que pára em respostas a todas as entradas como sendo a noção formal precisa correspondente à intuitiva idéia de algoritmo*
  - Tese de Church-Turing
- Máquina de Turing vista como um algoritmo
- O que é **efetivamente** computável?

Decidibilidade

- Antes de definir máquinas de Turing como programas (algoritmos) precisamos definir um “hardware” que execute tais programas
  - Máquinas de Turing Universais

Máquina de Turing Universal

- Uma Máquina de Turing que pode ser programada, podendo assim solucionar “qualquer” problema passível de resolução através de uma Máquina de Turing
- A Máquina de Turing Universal recebe como entrada uma máquina de Turing e a entrada a computar
- A entrada e a descrição da máquina são “programadas” em uma linguagem reconhecida pela Máquina de Turing Universal

Máquina de Turing Universal

- Convensão:
  - Um estado da máquina de Turing deverá ser da forma $\{q\}\{0,1\}^*$
  - Um símbolo de fita será representado como uma cadeia $\{a\}\{0,1\}^*$
  - Seja $M = (K, \Sigma, \delta, s, \{H\})$ uma Máquina de Turing
    - Sejam i e j os menores inteiros, tais que $2^i \geq |K|$ e $2^j \geq |\Sigma| + 2$ ($\leftarrow$ e $\rightarrow$)
      - Cada estado em K será representado como um símbolo q seguido de uma cadeia binária de comprimento i
      - Cada símbolo em Σ será representado como o símbolo a seguido de uma cadeia binária de comprimento j

Máquina de Turing Universal

- Convensão:
  - Para representar $\sqcup$, $\rhd$, $\rightarrow$ e $\leftarrow$ serão utilizados os quatro menores símbolos em ordem lexicográfica
    - $\sqcup$ - $a0^j$
    - $\rhd$ - $a0^{j-1}1$
    - $\leftarrow$ - $a0^{j-2}10$
    - $\rightarrow$ - $a0^{j-2}11$

Máquina de Turing Universal

- Convensão:
  - O estado inicial da Máquina será sempre representado como o primeiro estado lexicográfico $q0^i$
  - A tabela de transição σ da representação “M” da Máquina de Turing consiste em uma sequência de cadeias da forma

$$(q, a, p, b)$$

    onde q e p são representações de estados e a e b representações de símbolos

Máquina de Turing Universal

- Convensão:
  - As quádruplas são relacionadas em ordem lexicográfica crescente, começando com $\delta(s, \sqcup)$
  - O conjunto H é determinado indiretamente pela não ocorrência de seus estados como primeiros componentes em qualquer quádrupla de "M"
    - Se M decide uma linguagem ($H = \{q_{accept}, q_{reject}\}$), q_{accept} será, lexicograficamente, o menor dos dois estados de parada

Máquina de Turing Universal

- Qualquer Máquina de Turing pode ser representada segundo esssa convenção
- O mesmo método será utilizado para representar quaisquer cadeias sobre o alfabeto da máquina de Turing
  - para simplificar $\Gamma \subset \Sigma$
- a representação de w, será notacionada “w”

Máquina de Turing Universal

- Exemplo:
  - $M = (K, \Sigma, \delta, s, \{h\})$ onde:
    - $K = \{s, q, h\}$
    - $\Sigma = \{\sqcup, \triangleright, a\}$
    - δ :

estado	símbolo	δ
s	a	$(q, \sqcup)$
s	$\sqcup$	$(h, \sqcup)$
s	$\triangleright$	$(s, \rightarrow)$
q	a	(s, a)
q	$\sqcup$	$(s, \rightarrow)$
q	$\triangleright$	$(q, \rightarrow)$

Máquina de Turing Universal

- $|K| = 3$ então $i = 2$, pois $2^2 \geq 3$
- $|\Sigma| = 3$ então $j = 3$ pois $2^3 \geq 3 + 2$
- Estados e símbolos ficam assim representados

estado/símbolo	representação
s	$q00$
q	$q01$
h	$q11$
$\sqcup$	$a000$
$\triangleright$	$a001$
$\leftarrow$	$a010$
$\rightarrow$	$a011$
a	$a100$

Máquina de Turing Universal

- Exemplo:
  - A representação da cadeia $\triangleright aa \sqcup a$ será

$$"\triangleright aa \sqcup a" = a001a100a100a000a100$$

  - A representação “M” da máquina de Turing M será

$$\begin{aligned}"M" = \quad & (q00, a100, q01, a000), (q00, a000, q11, a000), \\ & (q00, a001, q00, a011), (q01, a100, q00, a011), \\ & (q01, a000, q00, a011), (q01, a001, q01, a011)\end{aligned}$$

Máquina de Turing Universal

- Apresentada a representação de qualquer máquina de Turing como "algoritmo", podemos agora introduzir uma Máquina de Turing Universal (U)
  - U recebe como entrada
    - uma representação "M" de uma certa máquina M
    - uma representação "w" de uma dada sentença de entrada w

Máquina de Turing Universal

- U pára em resposta a entrada “M”“w” se e somente se M para em resposta à entrada w

$$U(\text{''}M\text{''}, \text{''}w\text{''}) = \text{''}M(w)\text{''}$$

- A máquina U é uma MT de 3 fitas que opera da seguinte forma:
  - A cadeia “M”“w” é gravada em sua primeira fita
  - U move “M” para a segunda fita
  - U desloca “w” para a esquerda na primeira fita
  - U grava na terceira fita a codificação do estado inicial s de “M”, que é sempre $q0^i$ (os valore de i e j são obtidos por U via inspeção de “M”)

Máquina de Turing Universal

Continuação:

- U simula os passos de computação de M como segue:
  - percorre a segunda fita até encontrar uma quádrupla cujo primeiro componente corresponda ao estado codificado gravado na terceira fita e cujo segundo componente corresponda ao símbolo codificado apontado na primeira fita
  - Uma vez encontrada esta quádrupla, U altera o estado corrente, subtituindo-o pelo terceiro componente da quádrupla e realiza, em sua primeira fita, a ação especificada pelo quarto componente

Máquina de Turing Universal

Continuação:

- Se o quarto componente
  - codifica um símbolo do alfabeto da fita de M, esse símbolo é gravado na primeira fita, substituindo o símbolo corrente
  - for $a0^j10$ ($\longleftarrow$), U move o primeiro cabeçote para o primeiro símbolo a à esquerda
  - for $a0^j11$ ($\longrightarrow$), U move o primeiro cabeçote para o primeiro símbolo a à direita

Máquina de Turing Universal

Continuação:

- se um $\sqcup$ for encontrado, U o converte para a representação $a0^j$
- se em algum passo a combinação de estado/símbolo não for encontrada na segunda fita, isso significa que o estado corrente é um estado de parada
- U também pára em um estado de parada conveniente

O problema da Parada

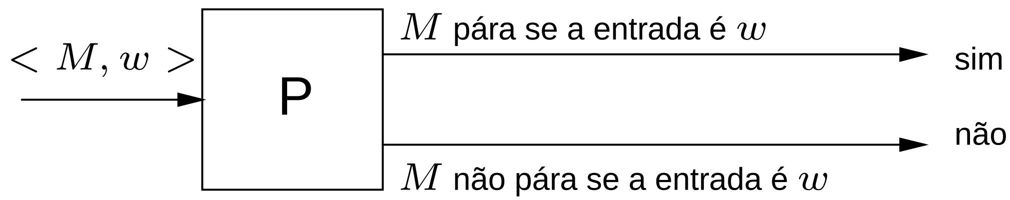
- Dadas uma Máquina de Turing arbitrária M e uma palavra arbitrária w, M pára ao computar w?

O problema da Parada

- 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 estrito da classe das linguagens recursivamente enumeráveis

O problema da Parada

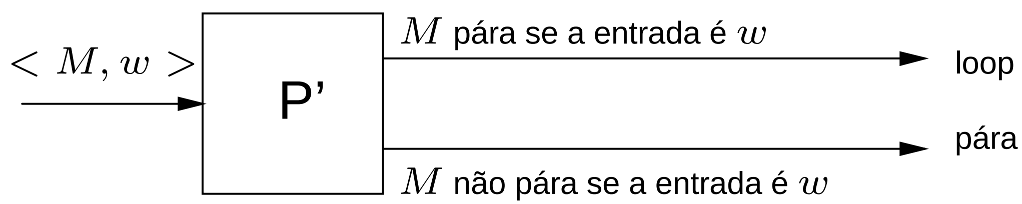
- Teorema: O problema da Parada para MT é indecidível.
- Prova: Suponha que o problema da parada seja decidível. Então existe uma MT P que sempre pára em resposta a uma entrada $< M, w >$



O problema da Parada

Continuação

- A partir da máquina P seria possível construir uma máquina P' que entra em loop se e somente se P pára em um estado final, ou ainda P' entra em loop se e somente se M pára com a entrada w



O problema da Parada

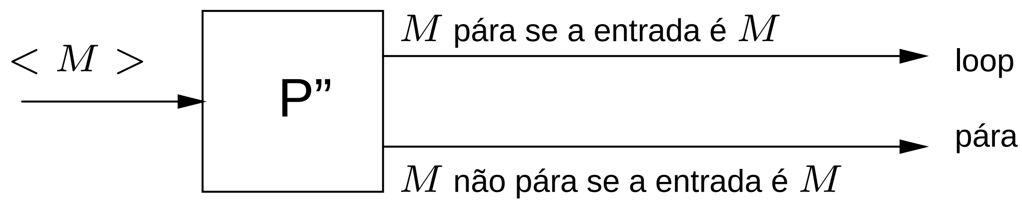
Continuação

- Para construir P', faça como segue:
  - para cada par (p, a), com $p \in K$ e $a \in \Gamma$ onde $\delta(p, a)$ é indefinida, $\delta(p, a) \rightarrow (l, a)$ onde l é um novo estado
  - crie novas transições $\delta(l, a) \rightarrow (l, a)$ para todo símbolo $a \in \Sigma$

O problema da Parada

Continuação

- A partir da máquina P' pode-se obter uma máquina P''



O problema da Parada

Continuação

- Para construir P'' a partir de P' construa transições para:
  - Duplicar a entrada $< M >$ para obter $< M, M >$ (basta copiar a entrada M na fita $M111M$)
  - Agir como P' sobre a entrada

O problema da Parada

Continuação

- agora considere o que acontece se $< P'' >$ for submetida como entrada para a MT P''
  - se P'' entra em loop, para a entrada $< P'' >$ é porque P'' pára se a entrada é $< P'' >$
  - se P'' pára quando a entrada é $< P'' >$ é porque P'' não pára se a entrada é $< P'' >$, ou seja

    P'' pára com entrada $< P'' >$ sse P'' não pára com entrada $< P'' >$

  - Mas P'' pode ser construída a partir de P. Assim, P não pode existir e portanto o problema da parada é indecidível

O problema da Parada

em Funções Recursivas Parciais

- O conjunto das funções recursivas parciais é *efetivamente* computável?
  - As funções recursivas parciais podem não ser definidas para todas as entradas (φ, ψ,ρ)
  - As funções recursivas totais são definidas para todas as entradas (f, g, h)
  - $\varphi(x)$ denota a aplicação da função (considerando-a como um procedimento) a x
    - não necessáriamente existe um objeto chamado $\varphi(x)$ pois φ aplicada a x pode ser indefinida

O problema da Parada

em Funções Recursivas Parciais

- Notacionaremos
  - $\varphi(x) \downarrow$ para "φ aplicada a x é definida"
  - $\varphi(x) \not\downarrow$ para "φ aplicada a x não é definida"

O problema da Parada

em Funções Recursivas Parciais

- Teorema: Não há função recursiva que decida se $\varphi(x)$ é definida
- Prova: Suponha que seja possível enumerar todas as funções recursivas parciais de uma variável por $\varphi_1, \varphi_2, ..., \varphi_n, ...$. Suponha que exista uma função recursiva total f, de modo que:

$$f(x) \simeq \begin{cases} 1 & \text{se } \varphi_x(x) \downarrow \\ 0 & \text{se } \varphi_x(x) \not\downarrow \end{cases}$$

O problema da Parada

em Funções Recursivas Parciais

- Então:

$$\rho(x) \simeq \begin{cases} \not\downarrow & \text{se } \varphi_x(x) \downarrow \\ 0 & \text{se } \varphi_x(x) \not\downarrow \end{cases}$$

é recursiva parcial, definida como segue:

$$\rho(x) \simeq \mu y(y + f(x) = 0)$$

Logo ρ deve ser φ_y para algum y.

O problema da Parada

em Funções Recursivas Parciais

- Mas então:

$$\rho(y) \simeq \begin{cases} \not\downarrow & \text{se } \varphi_y(y) \downarrow \\ 0 & \text{se } \varphi_y(y) \not\downarrow \end{cases}$$

O que é uma contradição. Logo não existe tal f.

O problema da Parada

em Funções Recursivas Parciais

- Tese de Church-Turing
  - Uma função é computável sse é recursiva parcial total.

O problema da Parada

- Considerações:
  - A linguagem $L(U)$ não é recursiva
  - Não há algoritmo que decida, para uma dada MT arbitrária M e uma palavra arbitrária de entrada w, se M aceita w ou não
    - Os problemas para os quais não existem algoritmos são ditos *indecidíveis*
  - O conjunto das linguagens recursivas é fechado em relação à operação de complemento
  - O conjunto das linguagens recursivamente enumeráveis não é fechada em relação à operação de complemento