

Lógica Computacional e Algoritmos

Flávio L. C. de Moura
Departamento de Ciência da Computação
Universidade de Brasília¹

19 de abril de 2022

¹flaviomoura@unb.br

Indecidibilidade da LPO

Nesta seção provaremos que o ganho de expressividade obtido ao passarmos da lógica proposicional para a lógica de primeira ordem vem com um custo. Sabemos que, dada uma fórmula φ da lógica proposicional, podemos (pelo menos teoricamente) decidir se φ é válida ou não: basta construirmos a tabela verdade de φ . Ainda que este processo seja ineficiente, já que cresce exponencialmente de acordo com o número de variáveis proposicionais que ocorrem em φ , ele nos fornece um algoritmo para decidir a validade na lógica proposicional. Esta ideia não pode ser utilizada na LPO, e como veremos o custo a ser pago com o ganho de expressividade é que a validade na LPO passa a ser um problema indecidível.

A prova da indecidibilidade da validade na LPO, também conhecido como *indecidibilidade da LPO*, será feita reduzindo o problema da validade na LPO a outro problema que já é conhecidamente indecidível. O problema indecidível que tomaremos é conhecido como o “problema da correspondência de Post (PCP)”: Dada uma sequência finita de pares $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ tal que todos os s_i 's e t_i 's são palavras binárias de comprimento positivo, existe uma sequência de índices i_1, i_2, \dots, i_n com $n \geq 1$ (que podem ser repetidos) tal que a concatenação das palavras $s_{i_1}s_{i_2}\dots s_{i_n}$ é igual a $t_{i_1}t_{i_2}\dots t_{i_n}$? Um exemplo de uma instância deste problema é $(1, 101), (10, 00), (011, 11)$. Note que esta instância possui a sequência $(1, 3, 2, 3)$ como solução. Isto significa que nosso espaço de busca é infinito, e isto nos dá uma certa intuição da razão pela qual este problema não é solúvel em geral.

Assumindo então a indecidibilidade do PCP, provaremos que a noção de validade na LPO também é indecidível conforme [1]. Iniciaremos definindo a redutibilidade entre problemas:

Definição 1. *Um problema A é redutível para um problema B se existe uma função (total) computável $f : A \rightarrow B$ tal que:*

$$x \in A \iff f(x) \in B.$$

Nossa prova consiste em reduzir o PCP para o problema da validade na LPO, construindo uma função computável que a cada instância do PCP retorna uma fórmula da LPO. Desta forma, se validade na LPO fosse decidível poderíamos construir um algoritmo para decidir PCP da seguinte forma:

- para cada instância C de PCP construímos uma fórmula ϕ_C tal que ϕ_C é válida se, e somente se, C possui uma solução.

Teorema 2. *A LPO é indecidível.*

Demonstração. A prova consiste em dada uma instância C do problema da correspondência de Post, construir em espaço e tempo finitos uma fórmula ϕ_C da LPO tal que $\models \phi_C$ se, e somente se, a instância C tem uma solução. Seja C a sequência $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$. Consideramos como símbolos de função e, f_0, f_1 de aridade 0, 1, 1 respectivamente. Interpretamos e como sendo a palavra vazia, e f_0 e f_1 como sendo a concatenação com 0 e 1, respectivamente. Assim, se $b_1b_2\dots b_l$ é uma palavra binária então

podemos codificá-la como sendo o termo $f_{b_1}(f_{b_2}(\dots(f_{b_m}(e))))$. Para facilitar a leitura das fórmulas abreviaremos um termo da forma $f_{b_1}(f_{b_2}(\dots(f_{b_m}(t))))$ por $f_{b_1 b_2 \dots b_m}(t)$. Também utilizaremos um predicado binário p , onde $p(s, t)$ significa “existe uma sequência de índices (i_1, i_2, \dots, i_m) tal que s é o termo representando $s_{i_1} s_{i_2} \dots s_{i_m}$ e t é o termo representando $t_{i_1} t_{i_2} \dots t_{i_m}$ ”. Ou seja, s constrói uma palavra utilizando a mesma sequência de índices utilizada por t . Nossa fórmula ϕ_C possui a seguinte estrutura: $(\phi_1 \wedge \phi_2) \rightarrow \phi_3$, onde:

$$\begin{aligned}\phi_1 &:= \bigwedge_{i=1}^k p(f_{s_i}(e), f_{t_i}(e)) \\ \phi_2 &:= \forall_v \forall_w (p(v, w) \rightarrow \bigwedge_{i=1}^k p(f_{s_i}(v), f_{t_i}(w))) \\ \phi_3 &:= \exists_z p(z, z)\end{aligned}$$

Afirmção: $\models \phi_C$ sse a instância C do problema da correspondência de Post tem uma solução.

Inicialmente vamos assumir que $\models \phi_C$. Nossa estratégia é encontrar um modelo \mathcal{M} para ϕ_C que nos diga que existe uma solução para a instância C por simples inspeção do significado da satisfatibilidade de ϕ_C para \mathcal{M} . O universo A de \mathcal{M} é o conjunto de todas as palavras binárias finitas incluindo a palavra vazia (que denotaremos por ϵ), isto é $A = \{0, 1\}^*$. A interpretação $e^{\mathcal{M}}$ da constante e é a palavra vazia ϵ . A interpretação $f_0^{\mathcal{M}}$ de f_0 é a concatenação de 0 a uma dada palavra: $f_0^{\mathcal{M}}(s) = s0$. Analogamente, $f_1^{\mathcal{M}}(s) = s1$. Por fim,

$$p^{\mathcal{M}} := \{(s, t) \mid \text{existe uma sequência de índices } (i_1, i_2, \dots, i_m) \text{ tais que } s \text{ é igual a } s_{i_1} s_{i_2} \dots s_{i_m} \text{ e } t \text{ é igual a } t_{i_1} t_{i_2} \dots t_{i_m}\}$$

onde s e t são palavras binárias e s_i e t_i são dados de C .

Por hipótese temos que $\models \phi$, e portanto $\mathcal{M} \models \phi$. Mostraremos que $\mathcal{M} \models \phi_3$, de onde podemos concluir que a instância C possui uma solução.

Afirmção 1: $\mathcal{M} \models \phi_1$. De fato, a sequência unitária (i) é tal que $p(f_{s_i}(e), f_{t_i}(e))$ é verdadeiro para todo $i = 1, \dots, k$.

Afirmção 2: $\mathcal{M} \models \phi_2$. De fato, $(s, t) \in p^{\mathcal{M}}$ implica que existe um sequência (i_1, i_2, \dots, i_m) tal que s é igual a $s_{i_1} s_{i_2} \dots s_{i_m}$ e t é igual a $t_{i_1} t_{i_2} \dots t_{i_m}$.

Escolhendo a sequência $(i_1, i_2, \dots, i_m, i)$ temos que ss_i é igual a $s_{i_1} s_{i_2} \dots s_{i_m} s_i$ e tt_i é igual a $t_{i_1} t_{i_2} \dots t_{i_m} t_i$, e portanto $\mathcal{M} \models \phi_2$. Temos que $\mathcal{M} \models (\phi_1 \wedge \phi_2) \rightarrow \phi_3$ e $\mathcal{M} \models \phi_1 \wedge \phi_2$, e portanto $\mathcal{M} \models \phi_3$. Pela definição de ϕ_3 e $p^{\mathcal{M}}$, concluímos que existe uma solução para C .

Reciprocamente, suponha que a instância C dada do problema da correspondência de Post possui uma solução, a saber (i_1, i_2, \dots, i_m) . Precisamos provar que se \mathcal{M}' é um modelo qualquer contendo a constante $e^{\mathcal{M}'}$, duas funções unárias $f_0^{\mathcal{M}'}$ e $f_1^{\mathcal{M}'}$, e um predicado binário $p^{\mathcal{M}'}$ então \mathcal{M}' satisfaz ϕ . Como ϕ é uma implicação, não há nada a fazer se $\mathcal{M}' \not\models \phi_1$ ou $\mathcal{M}' \not\models \phi_2$. A parte interessante é mostrar que $\mathcal{M}' \models \phi_3$ sempre que $\mathcal{M}' \models \phi_1 \wedge \phi_2$. Faremos isto interpretando palavras binárias finitas no domínio A' de \mathcal{M}' :

$$\begin{aligned}i(\epsilon) &:= e^{\mathcal{M}'} \\ i(s0) &:= f_0^{\mathcal{M}'}(i(s)) \\ i(s1) &:= f_1^{\mathcal{M}'}(i(s))\end{aligned}$$

Note que a função $i : \{0, 1\}^* \rightarrow A'$ é definida indutivamente sobre o comprimento de s .

A função de interpretação i aplica as funções $f_0^{\mathcal{M}'}$ e $f_1^{\mathcal{M}'}$ de forma reversa. Por exemplo, a palavra 0100110 é interpretada como $f_0^{\mathcal{M}'}(f_1^{\mathcal{M}'}(f_1^{\mathcal{M}'}(f_0^{\mathcal{M}'}(f_0^{\mathcal{M}'}(f_1^{\mathcal{M}'}(f_0^{\mathcal{M}'}(e^{\mathcal{M}'})\dots))))))$. Note que $i(b_1b_2\dots b_l) = f_{b_l}^{\mathcal{M}'}(f_{b_{l-1}}^{\mathcal{M}'}(\dots(f_{b_1}^{\mathcal{M}'}(e^{\mathcal{M}'})\dots)))$ corresponde ao significado dado para $f_s(e)$ em A' , onde s corresponde a $b_1b_2\dots b_l$. Sendo assim, e sabendo que $\mathcal{M}' \models \phi_1$ concluímos que $(i(s_i), i(t_i)) \in p^{\mathcal{M}'}$ para todo $i = 1, 2, \dots, k$. Analogamente, como $\mathcal{M}' \models \phi_2$ concluímos que para todo $(s, t) \in p^{\mathcal{M}'}$ temos que $(i(ss_i), i(tt_i)) \in p^{\mathcal{M}'}$ para todo $i = 1, 2, \dots, k$. Iniciando com $(s, t) = (s_{i_1}, t_{i_1})$ podemos utilizar o fato anterior repetidamente para obter que

$$(i(s_{i_1}s_{i_2}\dots s_{i_n}), i(t_{i_1}t_{i_2}\dots t_{i_n})) \in p^{\mathcal{M}'}$$

Como as palavras $s_{i_1}s_{i_2}\dots s_{i_n}$ e $t_{i_1}t_{i_2}\dots t_{i_n}$ formam uma solução de C estas palavras são iguais. Sendo assim, $i(s_{i_1}s_{i_2}\dots s_{i_n})$ e $i(t_{i_1}t_{i_2}\dots t_{i_n})$ representam o mesmo elemento de A' . Logo $\mathcal{M}' \models \exists_z p(z, z)$, isto é, $\mathcal{M}' \models \phi_3$. Logo validade na LPO é um problema indecidível. □

Referências Bibliográficas

- [1] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning About Systems*. Cambridge University Press, New York, NY, USA, 2004.