Introdução à Teoria da Computação

Exercícios

Livro: Michel Sipser, Introdução à Teoria da Computação – 2ª Ed.

Capítulo 02

EXERCÍCIOS

2.1 Retomemos a GLC G_4 que demos no Exemplo 2.4. Por conveniência, vamos renomear suas variáveis com apenas uma letra da seguinte forma.

$$
\begin{aligned}
E &\to E + T \mid T \\
T &\to T \times F \mid F \\
F &\to (E) \mid \mathtt{a}
\end{aligned}
$$

Dê árvores sintáticas e derivações para cada cadeia abaixo.

a. `a`
b. `a+a`
c. `a+a+a`
d. `((a))`

2.2 **a.** Use as linguagens $A = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^n \mid m, n \geq 0\}$ e $B = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^m \mid m, n \geq 0\}$ juntamente com o Exemplo 2.36 para mostrar que a classe das linguagens livres-do-contexto não é fechada sob interseção.

b. Use a parte (a) e a lei de DeMorgan (Teorema 0.20) para mostrar que a classe de linguagens livres-do-contexto não é fechada sob complementação.

R**2.3** Responda a cada item para a seguinte gramática livre-do-contexto G.

$$
\begin{aligned}
R &\to XRX \mid S \\
S &\to \mathtt{a}T\mathtt{b} \mid \mathtt{b}T\mathtt{a} \\
T &\to XTX \mid X \mid \varepsilon \\
X &\to \mathtt{a} \mid \mathtt{b}
\end{aligned}
$$

a. Quais são as variáveis de G?

b. Quais são os terminais de G?

c. Qual é a variável inicial de G?

d. Dê três cadeias em $L(G)$.

e. Dê três cadeias que *não* estão em $L(G)$.

f. Verdadeiro ou Falso: $T \Rightarrow \texttt{aba}$.

g. Verdadeiro ou Falso: $T \overset{*}{\Rightarrow} \texttt{aba}$.

h. Verdadeiro ou Falso: $T \Rightarrow T$.

i. Verdadeiro ou Falso: $T \overset{*}{\Rightarrow} T$.

j. Verdadeiro ou Falso: $XXX \overset{*}{\Rightarrow} \texttt{aba}$.

k. Verdadeiro ou Falso: $X \overset{*}{\Rightarrow} \texttt{aba}$.

l. Verdadeiro ou Falso: $T \overset{*}{\Rightarrow} XX$.

m. Verdadeiro ou Falso: $T \overset{*}{\Rightarrow} XXX$.

n. Verdadeiro ou Falso: $S \overset{*}{\Rightarrow} \varepsilon$.

o. Dê uma descrição em português de $L(G)$.

2.4 Dê gramáticas livres-do-contexto que gerem as seguintes linguagens. Em todos os itens o alfabeto Σ é $\{0,1\}$.

R**a.** $\{w|\ w$ contém pelo menos três 1s$\}$

b. $\{w|\ w$ começa e termina com o mesmo símbolo$\}$

c. $\{w|$ o comprimento de w é ímpar$\}$

R**d.** $\{w|$ o comprimento de w é ímpar e o símbolo do meio é um 0$\}$

e. $\{w|\ w = w^{\mathcal{R}}$, ou seja, w é um palíndromo$\}$

f. O conjunto vazio

2.5 Dê descrições informais e diagramas de estado de autômatos com pilha para as linguagens no Exercício 2.4.

2.6 Dê gramáticas livres-do-contexto gerando as seguintes linguagens.

R**a.** O conjunto de cadeias sobre o alfabeto $\{\texttt{a},\texttt{b}\}$ com mais as que bs

b. O complemento da linguagem $\{\texttt{a}^n\texttt{b}^n|\ n \geq 0\}$

R**c.** $\{w\texttt{\#}x|\ w^{\mathcal{R}}$ é uma subcadeia de x para $w, x \in \{0,1\}^*\}$

d. $\{x_1\texttt{\#}x_2\texttt{\#}\cdots\texttt{\#}x_k|\ k \geq 1$, cada $x_i \in \{\texttt{a},\texttt{b}\}^*$, e para algum i e j, $x_i = x_j^{\mathcal{R}}\}$

R**2.7** Dê descrições informais em português de APs para as linguagens no Exercício 2.6.

R**2.8** Mostre que a cadeia `the girl touches the boy with the flower` tem duas derivações mais à esquerda diferentes na gramática G_2 da página 105. Descreva em português os dois significados diferentes dessa sentença.

2.9 Dê uma gramática livre-do-contexto que gere a linguagem

$$A = \{\texttt{a}^i\texttt{b}^j\texttt{c}^k \mid i = j \text{ ou } j = k \text{ onde } i, j, k \geq 0\}.$$

Sua gramática é ambígua? Por que sim ou por que não?

2.10 Dê uma descrição informal de um autômato com pilha que reconheça a linguagem A do Exercício 2.9.

2.11 Converta a GLC G_4 dada no Exercício 2.1 para um AP equivalente, usando o procedimento dado no Teorema 2.20.

2.12 Converta a GLC G dada no Exercício 2.3 para um AP equivalente, usando o procedimento dado no Teorema 2.20.

2.13 Seja $G = (V, \Sigma, R, S)$ a seguinte gramática. $V = \{S, T, U\}$; $\Sigma = \{0, \texttt{\#}\}$; e R é o conjunto de regras:

$$S \to TT \mid U$$
$$T \to 0T \mid T0 \mid \texttt{\#}$$
$$U \to 0U00 \mid \texttt{\#}$$

a. Descreva $L(G)$ em português.

b. Prove que $L(G)$ não é regular.

2.14 Converta a seguinte GLC em uma GLC equivalente na forma normal de Chomsky, usando o procedimento dado no Teorema 2.9.

$$A \to BAB \mid B \mid \varepsilon$$
$$B \to 00 \mid \varepsilon$$

2.15 Dê um contra-exemplo para mostrar que a seguinte construção falha em provar que a classe das linguagens livres-do-contexto é fechada sob estrela. Seja A uma LLC que é gerada pela GLC $G = (V, \Sigma, R, S)$. Adicione a nova regra $S \to SS$ e chame a gramática resultante G'. Essa gramática é suposta gerar A^*.

2.16 Mostre que a classe de linguagens livres-do-contexto é fechada sob as operações regulares união, concatenação e estrela.

2.17 Use os resultados do Problema 2.16 para dar outra prova de que toda linguagem regular é livre-do-contexto, mostramos como converter uma expressão regular diretamente para uma gramática livre-do-contexto.

PROBLEMAS

R**2.18** **a.** Seja C uma linguagem livre-do-contexto e R uma linguagem regular. Prove que a linguagem $C \cap R$ é livre-do-contexto.

b. Use a parte (a) para mostrar que a linguagem $A = \{w \mid w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$ e contém o mesmo número de as, bs e cs$\}$ não é uma LLC.

***2.19** Suponha que a GLC G seja

$$S \to \mathtt{a}S\mathtt{b} \mid \mathtt{b}Y \mid Y\mathtt{a}$$
$$Y \to \mathtt{b}Y \mid \mathtt{a}Y \mid \varepsilon$$

Dê uma descrição simples de $L(G)$ em português. Use essa descrição para dar uma GLC para $\overline{L(G)}$, o complemento de $L(G)$.

2.20 Seja $A/B = \{w \mid wx \in A$ para algum $x \in B\}$. Mostre que, se A é livre-do-contexto e B é regular, então A/B é livre-do-contexto.

***2.21** Seja $\Sigma = \{\mathtt{a},\mathtt{b}\}$. Dê uma GLC que gera a linguagem das cadeias com duas vezes mais as que bs. Prove que sua gramática é correta.

***2.22** Seja $C = \{x\#y \mid x, y \in \{0,1\}^*$ e $x \neq y\}$. Mostre que C é uma linguagem livre-do-contexto.

***2.23** Seja $D = \{xy \mid x, y \in \{0,1\}^*$ e $|x| = |y|$ mas $x \neq y\}$. Mostre que D é uma linguagem livre-do-contexto.

***2.24** Seja $E = \{\mathtt{a}^i\mathtt{b}^j \mid i \neq j$ e $2i \neq j\}$. Mostre que E é uma linguagem livre-do-contexto.

2.25 Para qualquer linguagem A, seja $SUFIXO(A) = \{v \mid uv \in A$ para alguma cadeia $u\}$. Mostre que a classe de linguagens livres-do-contexto é fechada sob a operação $SUFIXO$.

2.26 Mostre que, se G for uma GLC na forma normal de Chomsky, então para qualquer cadeia $w \in L(G)$ de comprimento $n \geq 1$, exatamente $2n - 1$ passos são necessários para qualquer derivação de w.

***2.27** Seja $G = (V, \Sigma, R, \langle \text{STMT} \rangle)$ a seguinte gramática.

$$\begin{aligned}
\langle \text{STMT} \rangle &\to \langle \text{ASSIGN} \rangle \mid \langle \text{IF-THEN} \rangle \mid \langle \text{IF-THEN-ELSE} \rangle \\
\langle \text{IF-THEN} \rangle &\to \texttt{if condition then } \langle \text{STMT} \rangle \\
\langle \text{IF-THEN-ELSE} \rangle &\to \texttt{if condition then } \langle \text{STMT} \rangle \texttt{ else } \langle \text{STMT} \rangle \\
\langle \text{ASSIGN} \rangle &\to \texttt{a:=1}
\end{aligned}$$

$$\begin{aligned}
\Sigma &= \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{a:=1}\}. \\
V &= \{\langle \text{STMT} \rangle, \langle \text{IF-THEN} \rangle, \langle \text{IF-THEN-ELSE} \rangle, \langle \text{ASSIGN} \rangle\}
\end{aligned}$$

G é uma gramática aparentemente natural para um fragmento de uma linguagem de programação, mas G é ambígua.

a. Mostre que G é ambígua.

b. Dê uma nova gramática não-ambígua para a mesma linguagem.

***2.28** Dê GLCs não-ambíguas para as linguagens a seguir.

a. $\{w \mid$ em todo prefixo de w o número de as é pelo menos igual ao número de bs$\}$

b. $\{w \mid$ os números de as e de bs em w são iguais$\}$

c. $\{w \mid$ em w, o número de as é pelo menos igual ao número de bs$\}$

***2.29** Mostre que a linguagem A do Exercício 2.9 é inerentemente ambígua.

2.30 Use o lema do bombeamento para mostrar que as seguintes linguagens não são livres-do-contexto.

a. $\{0^n 1^n 0^n 1^n \mid n \geq 0\}$

R**b.** $\{0^n \# 0^{2n} \# 0^{3n} \mid n \geq 0\}$

R**c.** $\{w\#t \mid w$ é uma subcadeia de t, onde $w, t \in \{\texttt{a}, \texttt{b}\}^*\}$

d. $\{t_1 \# t_2 \# \cdots \# t_k \mid k \geq 2$, cada $t_i \in \{\texttt{a}, \texttt{b}\}^*$, e $t_i = t_j$ para algum $i \neq j\}$

2.31 Seja B a linguagem de todos os palíndromos sobre $\{0,1\}$ contendo o mesmo número de 0s e 1s. Mostre que B não é livre-do-contexto.

***2.32** Seja $\Sigma = \{1, 2, 3, 4\}$ e $C = \{w \in \Sigma^* \mid$ em w, o número de 1s é igual ao número de 2s, e o número de 3s é igual ao número de 4s$\}$. Mostre que C não é livre-do-contexto.

2.33 Mostre que $F = \{\texttt{a}^i \texttt{b}^j \mid i \neq kj$ para algum inteiro positivo $k\}$ não é livre-do-contexto.

2.34 Considere a linguagem $B = L(G)$, onde G é a gramática dada no Exercício 2.13. O lema do bombeamento para linguagens livres-do-contexto, Teorema 2.34, enuncia a existência de um comprimento de bombeamento p para B. Qual é o valor mínimo de p que funciona no lema do bombeamento? Justifique sua resposta.

2.35 Seja G uma GLC na forma normal de Chomsky que contém b variáveis. Mostre que, se G gera alguma cadeia com uma derivação tendo no mínimo 2^b passos, $L(G)$ é infinita.

2.36 Dê um exemplo de uma linguagem que não é livre-do-contexto, mas que age como uma LLC no lema do bombeamento. Prove que seu exemplo funciona. (Veja o exemplo análogo para linguagens regulares no Problema 1.54.)

*__2.37__ Prove a seguinte forma mais forte do lema do bombeamento, na qual *ambas* as partes v e y devem ser não-vazias quando a cadeia s é dividida.

Se A for uma linguagem livre-do-contexto, então existe um número k onde, se s é uma cadeia qualquer em A de comprimento no mínimo k, então s pode ser dividida em cinco partes, $s = uvxyz$, satisfazendo as condições:

a. para cada $i \geq 0$, $uv^ixy^iz \in A$,
b. $v \neq \varepsilon$ e $y \neq \varepsilon$, e
c. $|vxy| \leq k$.

R**2.38** Remeta-se ao Problema 1.41 para a definição da operação de embaralhamento perfeito. Mostre que a classe de linguagens livres-do-contexto não é fechada sob embaralhamento perfeito.

2.39 Remeta-se ao Problema 1.42 para a definição da operação de embaralhamento. Mostre que a classe de linguagens livres-do-contexto não é fechada sob embaralhamento.

*__2.40__ Digamos que uma linguagem é ***prefixo-fechada*** se o prefixo de qualquer cadeia na linguagem também está na linguagem. Seja C uma linguagem livre-do-contexto, infinita e prefixo-fechada. Mostre que C contém um subconjunto regular infinito.

*__2.41__ Leia as definições de NÃOPREFIXO(A) e NÃOESTENDE(A) no Problema 1.40.

a. Mostre que a classe de LLCs não é fechada sob a operação NÃOPREFIXO.
b. Mostre que a classe de LLCs não é fechada sob a operação NÃOESTENDE.

2.42 Seja $\Sigma = \{1, \#\}$ e $Y = \{w|\ w = t_1\#t_2\#\cdots\#t_k$ para $k \geq 0$, cada $t_i \in 1^*$, e $t_i \neq t_j$ sempre que $i \neq j\}$. Prove que Y não é livre-do-contexto.

2.43 Para cadeias w e t, escreva $w \doteq t$ se os símbolos de w são uma permutação dos símbolos de t. Em outras palavras, $w \doteq t$ se t e w têm os mesmos símbolos nas mesmas quantidades, mas possivelmente em uma ordem diferente.

Para qualquer cadeia w, defina $MISTURA(w) = \{t|\ t \doteq w\}$. Para qualquer linguagem A, seja $MISTURA(A) = \{t|\ t \in MISTURA(w)$ para alguma $w \in A\}$.

a. Mostre que, se $\Sigma = \{0,1\}$, então uma *MISTURA* de uma linguagem regular é livre-do-contexto.

b. O que acontece na parte (a) se Σ contém 3 ou mais símbolos? Prove sua resposta.

2.44 Se A e B são linguagens, defina $A \diamond B = \{xy|\ x \in A$ e $y \in B$ e $|x| = |y|\}$. Mostre que se A e B forem linguagens regulares, então $A \diamond B$ é uma LLC.

*__2.45__ Seja $A = \{wtw^{\mathcal{R}}|\ w, t \in \{0,1\}^*$ e $|w| = |t|\}$. Prove que A não é uma linguagem livre-do-contexto.

SOLUÇÕES SELECIONADAS

2.3 **(a)** R, X, S, T; **(b)** a, b; **(c)** R; **(d)** Três cadeias em G são ab, ba e aab; **(e)** Três cadeias que não estão em G são a, b e ε; **(f)** Falso; **(g)** Verdadeiro; **(h)** Falso; **(i)** Verdadeiro; **(j)** Verdadeiro; **(k)** Falso; **(l)** Verdadeiro; **(m)** Verdadeiro; **(n)** Falso; **(o)** $L(G)$ consiste de todas as cadeias sobre a e b que não são palíndromos.

2.4 **(a)** $S \to R1R1R1R$
$R \to 0R \mid 1R \mid \varepsilon$

(d) $S \to 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

2.6 **(a)** $S \to T\mathtt{a}T$
$T \to TT \mid \mathtt{a}T\mathtt{b} \mid \mathtt{b}T\mathtt{a} \mid \mathtt{a} \mid \varepsilon$

T gera todas as cadeias com pelo menos a mesma quantidade de as que bs, e S força um a extra.

(c) $S \to TX$
$T \to 0T0 \mid 1T1 \mid \#X$
$X \to 0X \mid 1X \mid \varepsilon$

2.7 **(a)** O AP usa sua pilha para contar o número de as menos o número de bs. Ele entra num estado de aceitação sempre que esse contador é positivo. Em mais detalhes, ele opera da seguinte maneira. O AP lê a entrada. Se ele vê um b e seu símbolo no topo da pilha é um a, ele desempilha. Similarmente, se ele lê um a e seu símbolo de topo de pilha é um b, ele desempilha. Em todos os outros casos, ele empilha o símbolo de entrada. Depois que o AP lê a entrada, se a estiver no topo da pilha, ele aceita. Caso contrário, ele rejeita.

(c) O AP faz uma varredura na cadeia de entrada e empilha todo símbolo que lê até que leia um #. Se # nunca for encontrado, ele rejeita. Então, o AP pula a parte da entrada, não-deterministicamente decidindo quando pára de pular. Nesse ponto, ele compara os próximos símbolos de entrada com os símbolos que ele desempilha. Em caso de qualquer desacordo, ou se a entrada terminar enquanto a pilha é não-vazia, esse ramo da computação rejeita. Se a pilha se torna vazia, a máquina lê o resto da entrada e aceita.

2.8 Aqui está uma derivação:

⟨SENTENCE⟩ ⇒ ⟨NOUN-PHRASE⟩⟨VERB-PHRASE⟩ ⇒
⟨CMPLX-NOUN⟩⟨VERB-PHRASE⟩ ⇒
⟨CMPLX-NOUN⟩⟨CMPLX-VERB⟩⟨PREP-PHRASE⟩ ⇒
⟨ARTICLE⟩⟨NOUN⟩⟨CMPLX-VERB⟩⟨PREP-PHRASE⟩ ⇒
The boy ⟨VERB⟩⟨NOUN-PHRASE⟩⟨PREP-PHRASE⟩ ⇒
The boy ⟨VERB⟩⟨NOUN-PHRASE⟩⟨PREP⟩⟨CMPLX-NOUN⟩ ⇒
The boy touches ⟨NOUN-PHRASE⟩⟨PREP⟩⟨CMPLX-NOUN⟩ ⇒
The boy touches ⟨CMPLX-NOUN⟩⟨PREP⟩⟨CMPLX-NOUN⟩ ⇒
The boy touches ⟨ARTICLE⟩⟨NOUN⟩⟨PREP⟩⟨CMPLX-NOUN⟩ ⇒
The boy touches the girl with ⟨CMPLX-NOUN⟩ ⇒
The boy touches the girl with ⟨ARTICLE⟩⟨NOUN⟩ ⇒
The boy touches the girl with the flower

Aqui está uma outra derivação:

⟨SENTENCE⟩ ⇒ ⟨NOUN-PHRASE⟩⟨VERB-PHRASE⟩ ⇒
⟨CMPLX-NOUN⟩⟨VERB-PHRASE⟩ ⇒ ⟨ARTICLE⟩⟨NOUN⟩⟨VERB-PHRASE⟩ ⇒
The boy ⟨VERB-PHRASE⟩ ⇒ The boy ⟨CMPLX-VERB⟩ ⇒
The boy ⟨VERB⟩⟨NOUN-PHRASE⟩ ⇒
The boy touches ⟨NOUN-PHRASE⟩ ⇒
The boy touches ⟨CMPLX-NOUN⟩⟨PREP-PHRASE⟩ ⇒

```
The boy touches ⟨ARTICLE⟩⟨NOUN⟩⟨PREP-PHRASE⟩ ⇒
The boy touches the girl ⟨PREP-PHRASE⟩ ⇒
The boy touches the girl ⟨PREP⟩⟨CMPLX-NOUN⟩ ⇒
The boy touches the girl with ⟨CMPLX-NOUN⟩ ⇒
The boy touches the girl with ⟨ARTICLE⟩⟨NOUN⟩ ⇒
The boy touches the girl with the flower
```

Cada uma dessas derivações corresponde a um significado diferente em inglês. Na primeira derivação, a sentença quer dizer que o garoto usou a flor para tocar a garota. Na segunda derivação, a garota está segurando a flor quando o garoto a toca.

2.18 **(a)** Seja C uma linguagem livre-do-contexto e R uma linguagem regular. Seja P o AP que reconhece C, e D o AFD que reconhece R. Se Q é o conjunto de estados de P e Q' é o conjunto de estados de D, construímos um AP P' que reconhece $C \cap R$ com o conjunto de estados $Q \times Q'$. P' fará o que P faz e também mantém registro dos estados de D. Ele aceita uma cadeia w se e somente se ele pára em um estado $q \in F_P \times F_D$, onde F_P é o conjunto de estados de aceitação de P e F_D é o conjunto de estados de aceitação de D. Como $C \cap R$ é reconhecida por P', ela é livre-do-contexto.

(b) Seja R a linguagem regular $\mathtt{a}^*\mathtt{b}^*\mathtt{c}^*$. Se A fosse uma LLC, então $A \cap R$ seria um LLC pela parte (a). No entanto, $A \cap R = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$, e o Exemplo 2.36 prova que $A \cap R$ não é livre-do-contexto. Por conseguinte, A não é uma LLC.

2.30 **(b)** Seja $B = \{\mathtt{0}^n\mathtt{\#0}^{2n}\mathtt{\#0}^{3n} \mid n \geq 0\}$. Seja p o comprimento de bombeamento dado pelo lema do bombeamento. Seja $s = \mathtt{0}^p\mathtt{\#0}^{2p}\mathtt{\#0}^{3p}$. Mostramos que $s = uvxyz$ não pode ser bombeada.

Nem v nem y pode conter #, caso contrário xv^2wy^2z contém mais que dois #s. Conseqüentemente, se dividirmos s em três segmentos por #s: $\mathtt{0}^p$, $\mathtt{0}^{2p}$ e $\mathtt{0}^{3p}$, pelo menos um dos segmentos não está contido em v ou y. Logo, xv^2wy^2z não está em B porque a proporção $1 : 2 : 3$ entre os comprimentos dos segmentos não é mantida.

(c) Seja $C = \{w\mathtt{\#}t \mid w \text{ é uma subcadeia de } t, \text{ onde } w, t \in \{\mathtt{a}, \mathtt{b}\}^*\}$. Seja p o comprimento de bombeamento dado pelo lema do bombeamento. Seja $s = \mathtt{a}^p\mathtt{b}^p\mathtt{\#a}^p\mathtt{b}^p$. Mostramos que a cadeia $s = uvxyz$ não pode ser bombeada.

Nem v nem y pode conter #, caso contrário uv^0xy^0z não contém # e, por conseguinte, não está em C. Se ambas v e y são não-vazias e ocorrem no lado esquerdo do #, a cadeia uv^2xy^2z não pode estar em C porque ela é mais longa no lado esquerdo do #. Similarmente, se ambas as cadeias ocorrem no lado direito do #, a cadeia uv^0xy^0z não pode estar em C porque ela é novamente mais longa no lado esquerdo do #. Se apenas uma das duas v e y é não-vazia (ambas não podem ser não-vazias), trate-as como se ambas ocorressem no mesmo lado do # como acima.

O único caso remanescente é aquele no qual ambas v e y são não-vazias e vão além do #. Mas, então v consiste de bs e y consiste de as devido à terceira condição do lema do bombeamento $|vxy| \leq p$. Logo, uv^2xy^2z contém mais bs no lado esquerdo do #, portanto ela não pode ser um membro de C.

2.38 Seja A a linguagem $\{\mathtt{0}^k\mathtt{1}^k \mid k \geq 0\}$ e suponha que B seja a linguagem $\{\mathtt{a}^k\mathtt{b}^{3k} \mid k \geq 0\}$. O embaralhamento perfeito de A e B é a linguagem $C = \{(\mathtt{0a})^k(\mathtt{0b})^k(\mathtt{1b})^{2k} \mid k \geq 0\}$. As linguagens A e B são facilmente vistas como sendo LLCs, mas C não é uma LLC, conforme o que segue. Se C fosse uma LLC, seja p o comprimento de bombeamento dado pelo lema do bombeamento, e suponha que

s seja a cadeia $(\texttt{0a})^p(\texttt{0b})^p(\texttt{1b})^{2p}$. Em razão de s ser mais longa que p e $s \in C$, podemos dividir $s = uvxyz$ satisfazendo as três condições do lema do bombeamento. As cadeias em C contêm duas vezes mais 1s que as. Para que uv^2xy^2z tenha essa propriedade, a cadeia vxy tem que conter tanto 1s quanto as. Mas isso é impossível, porque elas são separadas por $2p$ símbolos e a terceira condição diz que $|vxy| \leq p$. Logo, C não é livre-do-contexto.