MAC-414 – Tomatinhos — Prova 3 – 8/12/2009

Resoluções

1. Mostre que a gramática

$$S \to aSbS \mid bSaS \mid \lambda$$

gera a linguagem das palavras que tem o mesmo número de ocorrências de letras a e b.

Seja L a linguagem dada pela gramática e I a linguagem das palavras que tem o mesmo número de ocorrências de letras a e b.

Como cada produção introduz o mesmo número de a e b, segue fácil por indução que toda forma sentencial da gramática tem o mesmo número de ocorrências de letras a e b, logo $L \subseteq I$.

Agora, vamos mostrar, por indução, que se $x \in I$, então $x \in L$. A base é $x = \lambda$, claro. Vamos denotar $f(z) = |z|_a - |z|_b$, para cada palavra z; assim, $z \in I$ se e só se $f(z) = 0$.

Assim, dado $x \in I$, $x \neq \lambda$, seja w o menor prefixo não vazio de x que está em I; então $x = wu$, e claro que $u \in I$.

Pela escolha de w, $|u| < |x|$, logo, pela HI, $u \in L$. Vamos escrever $w = \sigma z \tau$, onde σ e τ são letras. Então,
$0 = f(w) = f(\sigma) + f(\tau) + f(z)$. Vamos considerar as quatro possibilidades para o par (σ, τ):

(a) $w = aza$. Nesse caso, $f(z) = -2$. Então, z tem um prefixo y tal que $f(y) = -1$, e ay é um prefixo de x em I, mais curto que w. Isso contradiz a escolha de w, portanto este caso não pode ocorrer.

(b) $w = bzb$. Análogo ao caso anterior.

(c) $w = azb$. Então, $f(z) = 0$, logo $z \in I$, e pela HI, $z \in L$. Então, temos derivações $S \Rightarrow^* z$, $S \Rightarrow^* u$, e podemos combiná-las em $S \Rightarrow aSbS \Rightarrow^* azbS \Rightarrow^* azbu = x$.

(d) $w = bza$. Análogo ao caso anterior.

2. Considere a linguagem $L = \{a^i b^j c^k \mid i, j, k \geq 0, i = j \text{ ou } j = k\}$.

(a) Dê uma gramática que gera L, com no máximo 6 não-terminais. Não se esqueça de mostrar, ainda que informalmente, que a gramática está correta.

$$\begin{aligned} S &\rightarrow TC \mid AW \\ A &\rightarrow aA \mid \lambda \\ C &\rightarrow cC \mid \lambda \\ T &\rightarrow aTb \mid \lambda \\ W &\rightarrow bWc \mid \lambda \end{aligned}$$

Para verificar a corretude, note que $L = L_1 \cup L_2$, onde $L_1 = \{a^i b^i c^k \mid i, k \geq 0\}$ e $L_2 = \{a^i b^k c^k \mid i, k \geq 0\}$. Já cansamos de ver que a linguagem gerada a partir de T é o velho AnBn, e a partir de C se gera c^*, logo, $S \Rightarrow TC \Rightarrow^* x$, se e só se $x \in L_1$. Analogamente, $S \Rightarrow AW \Rightarrow^* x$, se e só se $x \in L_2$.

(b) Mostre que a gramática da primeira parte é ambígua (existe um teorema que garante isso, mas o que eu quero é um exemplo explícito).

Abaixo estão duas derivações de abc, que claramente têm árvores distintas:

$S \Rightarrow TC \Rightarrow aTbC \Rightarrow abC \Rightarrow abcC \Rightarrow abc$

$S \Rightarrow AW \Rightarrow aAW \Rightarrow aW \Rightarrow abWc \Rightarrow abc$

3. Dê um algoritmo que, dada uma gramática livre de contexto, decide se ela gera a palavra λ (além de, possivelmente, outras). O algoritmo deve ser polinomial no tamanho da gramática. Para fins da análise, o tamanho de uma gramática é a soma dos comprimentos do lado direito das produções com o número de produções. Você pode supor que tanto terminais como não terminais são tipos simples; em particular, pode usar, se quiser, vetores indexados por terminais e não terminais.

   Em alto nível:

   Comece marcando todos os não-terminais A tais que existe uma produção $A \to \lambda$, deixando os outros desmarcados (rodada 1). Prossiga por rodadas. A cada rodada, examine os não-terminais desmarcados.

   Se, examinando A, encontra-se uma produção da forma $A \to w$, em que w é uma palavra formada só por não-terminais desmarcados, marca-se A.

   Condições de parada:

   - Se em algum momento marca-se S, retorna **sim**.
   - Se em alguma rodada ninguém é marcado, retorna **não**.

   Corretude: claro que se A é marcado, então $A \Rightarrow^* \lambda$. Agora, para cada não-terminal A tal que $A \Rightarrow^* \lambda$, seja $r(A)$ o comprimento da menor derivação dessas; claro que para qualquer outro não-terminal B que ocorre nessa derivação, $r(B) < r(A)$. Vamos mostrar que nesse caso, A é marcado numa rodada k, para algum $k \leq r(A)$. Isso porque uma tal derivação começa por $A \Rightarrow w$, onde w é formada só por não-terminais, e todos eles derivam λ, logo são marcados em rodadas de índice $< r(A)$.

   Tempo: seja n o número de não-terminais e N o tamanho da gramática. Claro que existem no máximo n rodadas, e, com uma estrutura de dados bem boba, cada rodada leva no máximo tempo N. Assim, o tempo total é $O(nN) = O(N^2)$, já que $n < N$.

   Baixando um pouco o nível, e implementando com um pouco de cuidado:

   > `marca` é um vetor booleano indexado por não-terminais
   >
   > num outro vetor indexado por não-terminais, cada posição vai conter uma lista de conjuntos: para cada produção $A \to w$, em que w é uma palavra formada só por não-terminais diferentes de A, adiciona-se à lista de A o conjunto dos não-terminais que ocorrem em w. Isso pode ser feito em tempo linear no tamanho da gramática. Um conjunto pode ser implementado por uma palavra em não-terminais.

   Outra forma de enunciar a mesma idéia:

   > Mantém um conjunto M, de não-terminais.
   >
   > para cada produção $A \to w$ tal que $A \notin M$
   >
   > > apague em w todos os não-terminais que estão em M
   > > se w virar λ, adicione A a M
   >
   > Organize rodadas e combine isso com as condições de parada acima

4. Mostre que $\{w\rho(w)w \mid w \in \Sigma^*\}$ é livre de contexto, se e só se $|\Sigma| = 1$. Só para lembrar: $\rho(w)$ é o reverso de w.

   Seja $L = \{w\rho(w)w \mid w \in \Sigma^*\}$. Se $|\Sigma| = 1$, $w = \rho(w)$ para toda w; se $\Sigma = \{a\}$, então $L = (aaa)^*$, que é regular, portanto LC.

   Se $|\Sigma| > 1$, sem perda de generalidade podemos supor que $\{a, b\} \subseteq \Sigma$. Aqui segue uma idéia geral para demonstrações de que L não é LC.

   Seja $H = L \cap R$, onde R é regular; se mostrarmos que H não é livre de contexto, segue que L também não é. Escolher bem R pode fazer uma grande diferença. Por exemplo, no caso, eu escolheria uma R que limitasse o número de alternâncias de a e b.

   Só com essa hipótese. suponha que H seja LC, e seja N dado pelo Lema do Bombeamento. Considere uma palavra $x \in H$ grande, e sejam v e t os trechos bombeáveis. Se algum deles contém um segmento ab ou ba, ao bombear resultam palavras em que o número de alternâncias de a e b é ilimitado, caindo fora de R, portanto fora de H. Assim, cada uma de v e t está em a^* ou b^*.

   Para completar a demonstração, é preciso especificar melhor R e depois x. Aqui vão dois exemplos:

   (a) $R = a^+b^+a^+b^+$.
   Antes de mais nada, note que se $w\rho(w)w \in R$, então necessariamente $w = a^ib^j$ para alguns i, j, logo $H = \{a^ib^{2j}a^{2i}b^j \mid i, j > 0\}$. Escolha $x = a^Nb^{2N}a^{2N}b^N = a^Nb^N\rho(a^Nb^N)a^Nb^N$; observe que v e t têm que estar em um segmento de comprimento $\leq N$, assim não podem ser constituídas da mesma letra. Pela simetria da situação, podemos supor que $v = a^i$, $t = b^j$, e pelo menos um entre i e j é positivo, digamos que $i > 0$. Ao bombear, um dos blocos de a cresce, e o outro não, e com isso se sai de H. Isto é uma contradição com a afirmativa de que v e t são trechos bombeáveis, logo H não é LC.

   (b) $R = ab^*aab^*aab^*a$.
   Numa palavra em H o w correspondente tem que ter exatamente 2 a's, portanto tem que ter a forma ab^na. Daí segue que $H = \{ab^naab^naab^na \mid n \geq 0\}$. Assim, trechos bombeáveis têm que estar em b^+. Mas ao bombear, pelo menos um bloco de b's cresce e no máximo dois, assim o resultado sai de H. Segue, novamente, que H não é LC.