

Übung zur Vorlesung

BERECHENBARKEIT UND KOMPLEXITÄT

Lösung Blatt 7

Tutoriumsaufgabe 7.1

Für eine gegebene CFG $G = (N, \Sigma, P, S)$ soll entschieden werden, ob $L(G)$ ein Palindrom enthält. Zeigen Sie, dass dieses Problem unentscheidbar ist.

Sie P die Sprache der CFG, wo $L(G)$ ein Palindrom enthält. Für einen Widerspruch nehmen wir an, dass P entscheidbar ist. Dann gibt es eine TM M mit $L(M) = P$. Wir zeigen $PCP \leq P$.

Konstruktion: Sei w eine Eingabe von PCP . Falls w nicht der Syntax vom PCP entspricht, bilde w auf nicht eine CFG $G \notin P$ ab. Sonst, kodiere w Dominos D_1, \dots, D_n mit Wörtern oben x_1, \dots, x_n und Wörtern unten y_1, \dots, y_n über Σ . Sei CFG $G = (\{S\}, \Sigma \cup \{\#\}, P, S)$, wobei $\# \notin \Sigma$ eine neues Symbol ist und P aus den Regeln

$$S \rightarrow x_1 S y_1^R \mid \dots \mid x_n S y_n^R \mid x_1 \# y_1 \mid \dots \mid x_n \# y_n,$$

besteht, wobei für $y = a_1 \dots a_m$ wir mit $y^R = a_m \dots a_1$ das gespiegelte Wort bezeichnen. Korrektheit:

(\Rightarrow) Sei $w \in PCP$. Dann kodiert w Dominos D_1, \dots, D_n mit Wörtern oben x_1, \dots, x_n und Wörtern unten y_1, \dots, y_n über Σ . Weiter gibt es eine endliche Folge von Dominos D_{i_1}, \dots, D_{i_s} so dass $x_{i_1} \dots x_{i_s} = y_{i_1} \dots y_{i_s}$. Für die konstruierte CFG G gibt es Ableitung

$$S \rightarrow x_{i_1} S y_{i_1}^R \rightarrow \dots \rightarrow x_{i_1} \dots x_{i_{s-1}} S y_{i_{s-1}}^R \dots y_{i_1}^R \rightarrow x_{i_s} \dots x_{i_s} \# y_{i_s}^R \dots y_{i_1}^R,$$

welches ein Palindrom ist. Also $G \in PCP$.

(\Leftarrow) Sei G eine konstruierte CFG, und $G \in P$. Jede Ableitung ist für ein $s \geq 1$ so, dass sie $(s-1)$ oft eine Regel $S \rightarrow x_i S y_i$ anwendet und abschließend eine Regel $S \rightarrow x_i y_i$. Da $G \in P$ gibt es mindestens eine Ableitung für ein Palindrom und diese hat die Form

$$S \rightarrow x_{i_1} S y_{i_1}^R \rightarrow \dots \rightarrow x_{i_1} \dots x_{i_{s-1}} S y_{i_{s-1}}^R \dots y_{i_1}^R \rightarrow x_{i_s} \dots x_{i_s} \# y_{i_s}^R \dots y_{i_1}^R,$$

für ein $s \geq 1$. Die ursprüngliche PCP-Instanz hat die Dominos D_1, \dots, D_n mit Wörtern oben x_1, \dots, x_n und Wörtern unten y_1, \dots, y_n . Da $x_{i_1} \dots x_{i_s} \# y_{i_s}^R \dots y_{i_1}^R$ ein Palindrom ist, gilt $y_{i_1} = y_{i_1}, \dots, x_{i_s} = y_{i_s}$. Daher hat die PCP-Instanz die Lösung D_{i_1}, \dots, D_{i_s} .

Tutoriumsaufgabe 7.2

Zeigen Sie, dass folgende arithmetische Befehle durch ein LOOP-Programm simuliert werden können:

(a) $x_i := x_j \div 1$ (modifizierte Vorgängerfunktion mit Ergebnis 0 falls $x_j = 0$)

```
 $x_i := 0;$   
 $y := 0;$   
LOOP  $x_j$  DO  
     $x_i := y;$   
     $y := y + 1$   
ENDLOOP
```

Jede Iteration der Schleife berechnet aus (x_i, y) die neuen Werte $(y, y + 1)$, d. h., es wird die Folge $(0, 0), (0, 1), (1, 2), \dots$ berechnet. Beginnend mit $(0, 0)$ liefern x_j Iterationen also das Paar $(x_j - 1, x_j)$.

(b) $x_i := x_j \div x_k$ (modifizierte Subtraktion mit Ergebnis 0 falls $x_j < x_k$)

Nach (a) ist die modifizierte Vorgängerfunktion berechenbar.

```
 $x_i := x_j;$   
LOOP  $x_k$  DO  
     $x_i := x_i \div 1$   
ENDLOOP
```

(c) $x_i := \min\{x_j, x_k\}$

Nach (b) ist die Subtraktion und nach Vorlesung das IF- $x_i = 0$ -THEN-ELSE-Konstrukt LOOP-berechenbar. Weiter gilt $x_j - x_k \leq 0$ gdw. $\min\{x_j, x_k\} = x_j$.

```
 $y := x_j \div x_k;$   
IF  $y = 0$  THEN  $x_i := x_j$  ELSE  $x_i := x_k$ 
```

Tutoriumsaufgabe 7.3

Ein LOOP-Z-Programm ist ein LOOP-Programm, das das LOOP-Konstrukt nicht verwendet. Es lässt sich zeigen, dass für jedes LOOP-Z-Programm P mit Variablen x_1, \dots, x_n natürliche Zahlen a_1, \dots, a_n und b existieren, so dass $f_P(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i + b$ gilt. Zeigen Sie: Es gibt kein LOOP-Z-Programm P , das die Funktion $x_1 x_2$ berechnet.

Es bezeichne $f(x_1, x_2)$ die Funktion, die von P berechnet wird, d. h., den Wert der ersten Variable nach Ausführung von P mit Werten x_1 und x_2 in den ersten beiden Variablen. Mit (a) folgt, dass natürliche Zahlen a_1, \dots, a_n und b existieren, sodass

$$f(x_1, x_2) \leq f_P(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i + b$$

für alle x_1, \dots, x_n gilt. Wähle x_1 als $\max\{3a_2, 3b\} + 1$, x_2 als $\max\{3a_1, 3b\} + 1$ und die restlichen x_i als 0. Dann gilt weiter

$$\sum_{i=1}^n a_i x_i + b < \frac{x_2}{3} x_1 + \frac{x_1}{3} x_2 + b < \frac{x_2}{3} x_1 + \frac{x_1}{3} x_2 + \frac{x_1 x_2}{3} = x_1 x_2.$$

Also gilt $f(x_1, x_2) < x_1 x_2$ für diese konkrete Wahl von Werten, d. h., P kann die Funktion $x_1 x_2$ nicht berechnen.

Optional, nur wenn am Ende noch Zeit ist: Beweis der Behauptung

Zeige via Induktion die allgemeinere Aussage, dass für $j \in \{1, \dots, n\}$ Werte $a_{j,1}, \dots, a_{j,n}$ und b_j existieren, sodass $\sum_{i=1}^n a_{j,i}x_i + b_j$ den Wert in der j -ten Variable beschreibt. Dann liefert $a_i := \sum_{j=1}^n a_{j,i}$ für $i \in \{1, \dots, n\}$ und $b := \sum_{j=1}^n b_j$ die gewünschten Werte.

Für den Induktionsanfang betrachte eine Zuweisung $x_s := x_t + c$. Wähle

$$a_{j,i} := \begin{cases} 1 & \text{falls } j = s \text{ und } i = t \\ 1 & \text{falls } j \neq s \text{ und } i = j \\ 0 & \text{sonst} \end{cases}$$

für $i, j \in \{0, \dots, n\}$ und

$$b_j := \begin{cases} c & \text{falls } j = s \\ 0 & \text{sonst} \end{cases}.$$

Dann gilt

$$[x_s := x_t + c](x_1, \dots, x_n)_j = \sum_{i=1}^n a_{j,i}x_i + b_j.$$

Für den Induktionsschritt betrachte Programme P_1 und P_2 , für die die Behauptung nach IV bereits gilt, d. h., für die die Werte $a_{j,i}^1$ und $b_{j,i}^1$ bzw. $a_{j,i}^2$ und $b_{j,i}^2$ existieren. Dann gilt

$$\begin{aligned} [P_1; P_2](x_1, \dots, x_n)_j &= [P_2]([P_1](x_1, \dots, x_n))_j \\ &= \sum_{i=1}^n a_{j,i}^2 \left(\sum_{i'=1}^n a_{i,i'}^1 x_{i'} + b_i^1 \right) + b_j^2 \\ &= \sum_{i'=1}^n \left(\sum_{i=1}^n a_{j,i}^2 \cdot a_{i,i'}^1 \right) x_{i'} + \sum_{i=1}^n a_{j,i}^2 \cdot b_i^1 + b_j^2, \end{aligned}$$

was die Aussage beweist.

Tutoriumsaufgabe 7.4

Beweisen Sie, dass die Wachstumsfunktion $F_P : \mathbb{N} \rightarrow \mathbb{N}$ des folgenden LOOP-Programms P die Beziehung $F_P(n) \in \Theta(n^3)$ erfüllt:

```

LOOP  $x_1$  DO
  LOOP  $x_2$  DO
    LOOP  $x_3$  DO  $x_4 := x_4 + 1$  ENDLOOP
  ENDLOOP
ENDLOOP

```

Bestimmen Sie weiterhin eine natürliche Zahl m_P , sodass $F_P(n) < A(m_P, n)$ für alle $n \in \mathbb{N}$ gilt.

P übersetzt den Eingabevektor (a_1, a_2, a_3, a_4) in den Ausgabevektor $(a_1, a_2, a_3, a_4 + a_1 \cdot a_2 \cdot a_3)$. Folglich gilt $f_P(a_1, a_2, a_3, a_4) = \prod_{i=1}^3 a_i + \sum_{i=1}^4 a_i$, was

$$F_P(n) = \max \left\{ \prod_{i=1}^3 a_i + \sum_{i=1}^4 a_i \mid a_1, \dots, a_4 \in \mathbb{N} \text{ mit } \sum_{i=1}^4 a_i \leq n \right\}$$

liefert. Damit gilt $F_P(n) \leq n^3 + 4 \cdot n$, also $F_P(n) \in \mathcal{O}(n^3)$. Für $n \in N$ ist eine der Zahlen $n, n-1, n-2$ durch drei teilbar und man erhält $F_P(n) \geq (\frac{n-2}{3})^3 + 3 \cdot \frac{n-2}{3}$, also $F_P(n) \in \Omega(n^3)$.

Um ein m_P zu bestimmen, verwende die „Regeln“ aus dem Induktionsbeweis aus der Vorlesung.

$$\left. \begin{array}{l} \text{LOOP } x_1 \text{ DO} \\ \quad \text{LOOP } x_2 \text{ DO} \\ \quad \quad \text{LOOP } x_3 \text{ DO} \\ \quad \quad \quad x_4 := x_4 + 1 \} < A(2, n) \\ \quad \quad \text{ENDLOOP} \\ \quad \text{ENDLOOP} \\ \text{ENDLOOP} \end{array} \right\} < A(3, n) \left\} < A(4, n) \left\} < A(5, n)$$

Also gilt $F_P(n) < A(5, n)$.