Postsches Korrespondenzproblem

Soeren Berken-Mersmann

DHBW Karlsruhe

17. April 2015

Gliederung

- Postsches Korrespondenzproblem
- 2 Simulation einer Turingmaschine
- 3 Beweis der Nichtberechenbarkeit des PKPs
- 4 Beweise für 2 Probleme der formalen Sprachen

Postsches Korrespondenzproblem

$$\begin{bmatrix} 1 \\ 111 \end{bmatrix} \begin{bmatrix} 10111 \\ 10 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

Wer findet eine Reihenfolge, so dass unten und oben jeweils die gleiche Folge steht?

Postsches Korrespondenzproblem

$$\begin{bmatrix} 1 \\ 111 \end{bmatrix} \begin{bmatrix} 10111 \\ 10 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

Wer findet eine Reihenfolge, so dass unten und oben jeweils die gleiche Folge steht?

$$I_1 = (2,1,1,3) : \begin{bmatrix} 10111 \\ 10 \end{bmatrix} \begin{bmatrix} 1 \\ 111 \end{bmatrix} \begin{bmatrix} 1 \\ 111 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} 001 \\ 0 \end{bmatrix} \begin{bmatrix} 01 \\ 011 \end{bmatrix} \begin{bmatrix} 01 \\ 101 \end{bmatrix} \begin{bmatrix} 10 \\ 001 \end{bmatrix}$$

Wer findet hierfür eine Lösung?

$$\begin{bmatrix} 001 \\ 0 \end{bmatrix} \begin{bmatrix} 01 \\ 011 \end{bmatrix} \begin{bmatrix} 01 \\ 101 \end{bmatrix} \begin{bmatrix} 10 \\ 001 \end{bmatrix}$$

Wer findet hierfür eine Lösung?

$$I_1 = (2,4,3,4,4,2,1,2,4,3,4,3,4,4,3,4,4,2,1,4,4,2,1,3,4,1,1,3,...)$$

$$\begin{bmatrix} 10\\101\end{bmatrix}\begin{bmatrix} 011\\11\end{bmatrix}\begin{bmatrix} 101\\011\end{bmatrix}$$

$$\begin{bmatrix} 10\\101 \end{bmatrix} \begin{bmatrix} 011\\11 \end{bmatrix} \begin{bmatrix} 101\\011 \end{bmatrix}$$
$$\begin{bmatrix} 10\\101 \end{bmatrix}$$

$$\begin{bmatrix} 10\\101 \end{bmatrix} \begin{bmatrix} 011\\11 \end{bmatrix} \begin{bmatrix} 101\\011 \end{bmatrix}$$
$$\begin{bmatrix} 10\\101 \end{bmatrix} \begin{bmatrix} 101\\011 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ 101 \end{bmatrix} \begin{bmatrix} 011 \\ 11 \end{bmatrix} \begin{bmatrix} 101 \\ 011 \end{bmatrix}$$
$$\begin{bmatrix} 10 \\ 101 \end{bmatrix} \begin{bmatrix} 101 \\ 011 \end{bmatrix} \begin{bmatrix} 101 \\ 011 \end{bmatrix} ...$$

Dieses mal offensichtlich ohne Lösung

Postsches Korrespondenzproblem (formell)

Definition des PKP

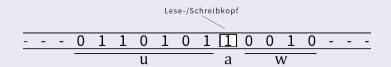
Gegeben sei eine endliche Menge an Wortpaaren $K=((x_1,y_1),...,(x_k,y_k))$, über dem Alphabet Σ mit $x_i,y_i\in\Sigma$. Gibt es eine Folge von Indizes $i_1,i_2,...,i_n\in 1,2,...,k,n\geq 1$, so dass $x_{i_1},x_{i_2},...x_{i_n}=y_{i_1},y_{i_2},...,y_{i_n}$.

Simulation einer Turingmaschine

Um zu beweisen, dass das PKP nicht berechenbar ist, werden wir eine Turingmaschine simulieren.

Dafür müssen wir zuerst den Rechenweg einer Turingmaschine formalisieren.

Zustand einer Turingmaschine



- Linkskontext: u
- Interner Zustand: *q*
- Gelesenes Symbol: a
- Rechtskontext: w

Somit lässt sich der Zustand Q_t einer Turingmaschine zum Zeitpunkt t durch die Folge $u_tq_ta_tw_t$ darstellen.



Rechenweg

Den Rechenweg einer Turingmaschine können wir als die Folge von Zuständen $Q_0, ..., Q_n$ vom Startzeitpunkt t=0 bis zum Endzeitpunkt t=n bei dem die Turingmaschine einen der Endzustände erreicht hat.

Formalisierte Darstellung: $0110101q_0100010\sharp 011010111q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Formalisierte Darstellung: $0110101q_0100010\sharp 011010111q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 0110101q_0100010\sharp$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 0$ $0110101q_0100010\sharp 0$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 01$ $0110101q_0100010\sharp 01$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 011010111q_100010\sharp 011$ $0110101q_0100010\sharp 011$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 0110$ $0110101q_0100010\sharp 0110$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp01101011q_100010\sharp01101$ $0110101q_0100010\sharp01101$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 011010$ $0110101q_0100010\sharp 011010$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp01101011q_100010\sharp0110101$ $0110101q_0100010\sharp0110101$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 01101011$ $0110101q_0100010\sharp 01101011$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

0110101 q_0 100010 \sharp 01101011 q_1 00010 \sharp 011010111 q_1 0110101 q_0 100010 \sharp 01101011 q_1 0

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 0110101111q_10$ $0110101q_0100010\sharp 01101011q_100$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

 $0110101q_0100010\sharp 01101011q_100010\sharp 011010111q_10010\\0110101q_0100010\sharp 01101011q_100010$

Formalisierte Darstellung: $0110101q_0100010\sharp 01101011q_100010$ Der Lesekopf liest eine 1 und befindet sich in Zustand q_0 , die Regel die Anwendung gefunden hat ist $q_01 \to q_11R$.

Simulation der Regel $q_10 o q_11R$

0110101 q_0 100010 \sharp 01101011 q_1 00010 \sharp 011010111 q_1 0010 \sharp 0110101 q_0 100010 \sharp 01101011 q_1 00010 \sharp

1. Anfangsregel

 $(\sharp\sharp q_0w\sharp,\sharp)$

1. Anfangsregel

 $(\sharp\sharp q_0w\sharp,\sharp)$

2. Kopierregeln

(a, a) für alle $a \in \Gamma \cup \{\sharp\}$

1. Anfangsregel

$$(\sharp\sharp q_0w\sharp,\sharp)$$

2. Kopierregeln

(a, a) für alle
$$a \in \Gamma \cup \{\sharp\}$$

3. Überführungsregeln

$$(cq', qa)$$
 falls $qa \rightarrow q'cR$, für $q \in Q$, $a \in \Gamma$
 $(q'bc, bqa)$ falls $qa \rightarrow q'cL$, für $q \in Q$, $a \in \Gamma$

1. Anfangsregel

```
(\sharp\sharp q_0w\sharp,\sharp)
```

2. Kopierregeln

(a, a) für alle
$$a \in \Gamma \cup \{\sharp\}$$

3. Überführungsregeln

$$(cq',qa)$$
 falls $qa \rightarrow q'cR$, für $q \in Q$, $a \in \Gamma$
 $(q'bc,bqa)$ falls $qa \rightarrow q'cL$, für $q \in Q$, $a \in \Gamma$

4. Aufholregeln

1. Anfangsregel

 $(\sharp\sharp q_0w\sharp,\sharp)$

2. Kopierregeln

(a, a) für alle
$$a \in \Gamma \cup \{\sharp\}$$

3. Überführungsregeln

$$\begin{array}{ll} (cq',qa) & \text{falls } qa \to q'cR, \text{ für } q \in Q, a \in \Gamma \\ (q'bc,bqa) & \text{falls } qa \to q'cL, \text{ für } q \in Q, a \in \Gamma \end{array}$$

4. Aufholregeln

$$(q, aq)$$
 für $a \in \Gamma$ und $q \in Q_f$
 (q, qa) für $a \in \Gamma$ und $q \in Q_f$

5. Abschlussregel

$$(\sharp, g\sharp\sharp)$$
 für $g\in Q_f$

Aufhol- und Abschlussregeln

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufhol- und Abschlussregeln

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0110101q_f10010\sharp$

Aufhol- und Abschlussregeln

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0110101q_f10010\sharp 0$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0110101q_f10010\sharp011010$ 011010

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0110101q_f10010\sharp 011010q_f$ $0110101q_f$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

0110101 q_f 10010 \sharp 011010 q_f 1
0110101 q_f 1

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

0110101 q_f 10010 \sharp 011010 q_f 10010 \sharp 0110101 q_f 10010 \sharp

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

0110101 q_f 10010 \sharp 011010 q_f 10010 \sharp 01101 q_f 10010 \sharp 0110101 q_f 10010 \sharp

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

0110101 q_f 10010 \sharp 011010 q_f 10010 \sharp 01101 q_f 10010 \sharp 01101 q_f 10010 \sharp 0110101 q_f 10010 \sharp 0110101 q_f 10010 \sharp

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp 0q_f 10010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp 0q_f 10010 \sharp q_f 10010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp 0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp 0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp 0q_f 010 \sharp q_f 010 \sharp 0q_f 010$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp q_f 0 \sharp 0010 \sharp q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp 0010 \sharp q_f 0010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp q_f 0 \sharp q_f \sharp 0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp$

Aktuelle Situation

Aktueller Zustand der Turingmaschine: $0110101q_f10010$ Die Turingmaschine ist in einem akzeptierenden Zustand q_f angekommen.

Aufholen und Abschließen des PKPs

 $0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp q_f 0 \sharp q_f \sharp \sharp \\ 0q_f 10010 \sharp q_f 10010 \sharp q_f 0010 \sharp q_f 010 \sharp q_f 10 \sharp q_f 0 \sharp q_f \sharp \sharp$

Beweis der Nichtberechenbarkeit

Beweis der Nichtberechenbarkeit

Reduktion des Halteproblems auf das PKP:

Sei M eine Turingmaschine und w ihre Eingabe, so lässt sich das Halteproblem durch die Übergangsfunktion f auf das PKP reduzieren:

$$X_{Halte}(M, w) \Leftrightarrow X_{PKP}(f(M, w))$$

Beweise weiterer Probleme

Seien G_1 und G_2 zwei kontextfreie Grammatiken, und $L_1 = L(G_1)$ und $L_2 = L(G_2)$ zwei daraus konstruierte kontextfreie Sprachen.

- 1 Ist G_1 eindeutig? (Mehrdeutigkeitstest)
- 2 Ist $L_1 = L_2$? (Äquivalenz)

Mehrdeutigkeitstest

Reduktion des PKPs auf den Mehrdeutigkeitstest

Gegeben eine Instanz des PKPs mit $\{(x_1, y_1), ..., (x_k, y_k)\}$ über einem endlichen Alphabet Σ und $I = \{a_1, ..., a_k\} \notin \Sigma$.

Wir konstruieren eine kontextfreie Grammatik

$$G_x = (\{S_x\}, T, P_x, S_x)$$
 mit $T = \Sigma \cup I$ und den Produktionen

$$P_{x} = \{S_{x} \to x_{1}S_{x}a_{1}|...|x_{n}S_{x}a_{n}|x_{1}a_{1}|...|x_{n}a_{n}\}.$$

Zusätzlich konstruieren wir eine zweite Grammatik G_y analog und eine weitere kfG G mit den Produktionsregeln

$$P = (S \to S_x | S_y) \cup P_x \cup P_y.$$



- G_X und G_Y sind offensichtlich eindeutig
- $L(G_x)$ erzeugt alle Wörter $x_{i_k}...x_{i_1}a_{i_1}...a_{i_k}$ und $L(G_y)$ analog $y_{i_k}...y_{i_1}a_{i_1}...a_{i_k}$.
- G ist dann mehrdeutig, wenn G_X und G_Y mindestens ein gemeinsames Wort erzeugen.
- Dann hat das PKP $x_{i_k}...x_{i_1} = y_{i_k}...y_{i_1}$ mindestens eine Lösung mit der Indexfolge $I = i_1, i_2, ...i_k$.
- Somit lässt sich das PKP auf den Mehrdeutigkeitstest reduzieren, und der Mehrdeutigkeitstest ist folglich nicht berechenbar.

Äquivalenz

Mit dem Beweis haben wir bereits gezeigt, dass $L(G_x) \cap L(G_y) = \emptyset$ nicht berechenbar ist.

Ferner lässt sich feststellen, dass G_x , G_y sogar deterministisch kontextfrei sind.

DkfG sind unter Komplementbildung abgeschlossen.

kfG sind unter Vereinigung abgeschlossen.

$$(G_1, G_2) \in \mathsf{Schnittproblem} \Leftrightarrow L(G_1) \cap L(G_2) = \emptyset$$

$$(G_1, G_2) \in \mathsf{Schnittproblem} \Leftrightarrow \mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$$

 $\Leftrightarrow \mathcal{L}(G_1) \subseteq \overline{\mathcal{L}(G_2)}$

$$(G_1, G_2) \in \mathsf{Schnittproblem} \Leftrightarrow \mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset \\ \Leftrightarrow \mathcal{L}(G_1) \subseteq \overline{\mathcal{L}(G_2)} \\ \Leftrightarrow \mathcal{L}(G_1) \subseteq \mathcal{L}(G_2')$$

$$(G_1, G_2) \in \mathsf{Schnittproblem} \Leftrightarrow \mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$$

 $\Leftrightarrow \mathcal{L}(G_1) \subseteq \overline{\mathcal{L}(G_2)}$
 $\Leftrightarrow \mathcal{L}(G_1) \subseteq \mathcal{L}(G_2')$
 $\Leftrightarrow \mathcal{L}(G_1) \cup \mathcal{L}(G_2') = \mathcal{L}(G_2')$

$$(G_1, G_2) \in \mathsf{Schnittproblem} \Leftrightarrow \mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$$
 $\Leftrightarrow \mathcal{L}(G_1) \subseteq \overline{\mathcal{L}(G_2)}$
 $\Leftrightarrow \mathcal{L}(G_1) \subseteq \mathcal{L}(G_2')$
 $\Leftrightarrow \mathcal{L}(G_1) \cup \mathcal{L}(G_2') = \mathcal{L}(G_2')$
 $\Leftrightarrow \mathcal{L}(G_3) = \mathcal{L}(G_2')$



$$(G_1,G_2) \in \mathsf{Schnittproblem} \Leftrightarrow \mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset \ \Leftrightarrow \mathcal{L}(G_1) \subseteq \overline{\mathcal{L}(G_2)} \ \Leftrightarrow \mathcal{L}(G_1) \subseteq \mathcal{L}(G_2') \ \Leftrightarrow \mathcal{L}(G_1) \cup \mathcal{L}(G_2') = \mathcal{L}(G_2') \ \Leftrightarrow \mathcal{L}(G_3) = \mathcal{L}(G_2') \ \Leftrightarrow \mathcal{L}(G_3,G_2') \in \mathsf{Äquivalenzproblem}$$



Quellenangaben

- Wegener, Ingo: Theoretische Informatik eine algorithmenorientierte Einführung. Dritte Auflage. Teubner, 2005. ISBN 3-8351-0033-5
- Schöning, Uwe: Theoretische Informatik kurz gefasst. Fünfte Auflage. Spektrum Akademischer Verlag, Heidelberg 2008, ISBN 978-3827418241

Vielen Dank für Ihre Aufmerksamkeit!