

Übung zur Vorlesung Automaten, Formale Sprachen und Berechenbarkeit

Hinweis: Die Zentralübung zur Vorlesung "Automaten, Formale Sprachen und Berechenbarkeit" findet jeden

Mittwoch zwischen 16:00 und 17:30 Uhr im Raum 03.09.014

statt. Die Übungsblätter werden jeweils in der Zentralübung am Mittwoch ausgegeben bzw. online gestellt und werden in der Übungsstunde der darauffolgenden Woche besprochen.

Informationen zur Vorlesung und Übung, insbesondere Lösungsvorschläge zu ausgewählten Übungsaufgaben, werden im WWW unter

<http://www.model.in.tum.de/courses/AFSB1Ex>

ausgehängt. Am Ende des Semesters finden mündliche Prüfungen statt. Weitere Einzelheiten werden rechtzeitig in Übung und Vorlesung bekannt gegeben. Bei erfolgreicher Teilnahme wird ein Übungsschein „Automaten, Formale Sprachen und Berechenbarkeit“ ausgestellt. Sollten Sie weitere organisatorische Fragen oder Fragen zum Stoff der Vorlesung haben, so wenden Sie sich bitte an Herrn Holzer, Raum MI 03.11.057, Tel. 289-17230, holzer@in.tum.de oder Herrn Tautschnig, Raum MI 03.11.055, Tel. 289-17232, tautschn@in.tum.de.

Wir wünschen Ihnen für das Wintersemester 2007/08 viel Erfolg.

Aufgabe 1 Wiederholung

Wiederholen Sie die Kapitel Automaten, Formale Sprachen und Berechenbarkeit aus Ihren Unterlagen zu Einführung in die Informatik 4.

Aufgabe 2 Kleensche Hülle

Seien Σ ein Alphabet und $A, B \subseteq \Sigma^*$. Geben Sie an, ob die folgenden Aussagen wahr oder falsch sind:

- (i) $|A| < |A^*|$
- (ii) $A \subseteq B$ impliziert $A^* \subseteq B^*$
- (iii) $(A \cap B)^* = A^* \cap B^*$
- (iv) $(A \cup B)^* = A^* \cup B^*$
- (v) $(AB)^* = A^*B^*$

Aufgabe 3 *uvw-Theorem*

Sei

$$L = \{wc \mid w \in \{a, b, c\}^* \text{ und } |w|_a < |w|_b\},$$

wobei $|w|_a$ die Anzahl der a -Symbole in w angibt (analog für $|w|_b$). D.h., die Sprache L besteht aus allen Wörtern über dem Alphabet $\{a, b, c\}$, in denen weniger a -Symbole als b -Symbole auftreten, gefolgt von einem Endsymbol c . Zeigen Sie mit dem *uvw*-Theorem dass L nicht regulär ist.

Aufgabe 4 *WHILE-Programme*

Ein WHILE-Programm ist ein Wort über dem Zeichenvorrat

- x_1, x_2, x_3 , usw. ... Variablen
- 0 ... Null
- succ, pred ... Nachfolge und Vorgänger
- $(,)$... Klammer
- $:=$... Zuweisung
- $;$... Hintereinanderausführung
- $\text{while}, \neq, \text{do}, \text{od}$... WHILE-Anweisung

Die Menge der WHILE-Programme hat folgende Syntax:

$\langle \text{Prog} \rangle ::= \langle \text{Zuweisung} \rangle \mid \langle \text{Prog} \rangle ';' \langle \text{Prog} \rangle \mid \text{'while'} \langle \text{Var} \rangle \text{'\neq'} \langle \text{Var} \rangle \text{'do'} \langle \text{Prog} \rangle \text{'od'}$

$\langle \text{Zuweisung} \rangle ::= \langle \text{Var} \rangle \text{' := 0' } \mid \langle \text{Var} \rangle \text{' := succ('} \langle \text{Var} \rangle \text{')' } \mid \langle \text{Var} \rangle \text{' := pred('} \langle \text{Var} \rangle \text{')' }$

Die Semantik stimmt mit der intuitiven Interpretation der Ausdrücke überein.

- (i) Schreiben Sie ein WHILE-Programm (ohne Makros), das die Funktion

$$f(x) = 2x$$

berechnet.

- (ii) Zeigen Sie, dass es kein WHILE-Programm mit nur einer Variable gibt, das f berechnet.