Aufgabe 36. Sei eine Funktion f definiert als

$$f(n) = \begin{cases} 3n+1 & \text{falls } n \text{ ungerade} \\ \frac{n}{2} & \text{falls } n \text{ gerade} \end{cases}$$

Man kann sich nun folgende Prozedur vorstellen. Man wähle eine positive natürliche Zahl und wende f darauf an. Ist das Ergebnis gleich 1, so stoppe man, anderenfalls, wende man wieder f auf das Ergebnis an und führe das Verfahren solange fort, bis man eine 1 als Ergebnis erhält.
Sei ν eine Funktion, die als Input positive natürliche Zahlen akzeptiert und als Ausgabe die Anzahl der Iterationen ausgibt, die im obigen Verfahren notwendig sind, um als Ergebnis für f eine 1 zu erhalten.
Zeigen Sie dass, ν eine rekursive Funktion ist. Sie können dabei alle Sätze aus dem Skriptum verwenden.
Sehr schwierige Zusatzaufgabe: Ist ν sogar primitiv rekursiv?

Aufgabe 37. Skizzieren Sie einen Beweis dafür, daß jede rekursive Funktion mit einem RAM Programm berechnet werden kann.
(Hinweis: Zeigen Sie, daß die Grundfunktionen berechnet werden können, und zeigen Sie, daß mit gegebenen berechenbaren Funktionen auch deren Komposition, Rekursion, und Minimalisierung berechnet werden kann.)

Aufgabe 38. Sei $L = \left\{ ww^{-1} \,\middle|\, w \in \{1,2\}^* \right\}$. Geben Sie eine möglichst gute Abschätzung der (worst-case) Zeit- und Raumkomplexität einer RAM, die L akzeptiert.

Aufgabe 39. Sei $L = \left\{ ww^{-1} \,\middle|\, w \in \{1,2\}^* \right\}$. Geben Sie eine möglichst gute Abschätzung der (worst-case) Zeit- und Raumkomplexität einer Turingmaschine, die L akzeptiert.

Aufgabe 40. Analog zur Diagonalsprache (siehe Skriptum Definition 4.1.3) kann man noch unendlich viele weitere Sprachen konstruieren, die alle nicht rekursiv aufzählbar sind.
Geben Sie wenigstens zwei weitere derartige Sprachen an und zeigen sie, daß diese Sprachen nicht rekursiv aufzählbar sind.