

Fragen, Antworten, Kommentare und Hinweise

Das Video zur Lösung der Aufgabe 8a finden Sie unter: <https://youtu.be/1qx6II810Ng>

Das Video zur Lösung der Aufgabe 8b finden Sie unter: <https://youtu.be/umY2HsIf48I>

für $n=0$ ist der Start nicht ok, korrigiert (erste Zeile streichen):

Start # suche0 # L

suche0 c sucheb # L

suche0 # S # S

weiterhin fehlen:

sucheb B sucheb B L

neuesc B neuesc B R

neuesB A neuesB A R

Frage: Wie gehören die Begriffe Entscheidbarkeit, Aufzählbarkeit und Akzeptierbarkeit zusammen?

Akzeptierbarkeit einer Sprache L ist zunächst die Forderung, dass es eine Turing-Maschine gibt, die irgendwann anhält, wenn sie auf ein Wort der Sprache L angesetzt wird.

Dies ist eng verwandt mit dem Begriff der Aufzählbarkeit, die nur die zusätzliche Form der Ergebnisausgabe der Turing-Maschine fordert. Wenn es für L eine akzeptierende Turing-Maschine gibt, ist es trivial daraus eine aufzählende Turing-Maschine konstruieren (erste Diagonalisierungsidee, aller Worte schrittweise ausprobieren). Genauso kann aus der aufzählenden Turing-Maschine eine akzeptierende gebaut werden, es werden mit der gleichen Diagonalisierungsidee alle Turing-Maschinen mit l^n als Eingabe ausgeführt und angehalten, wenn das Wort gefunden wurde. Etwas unsauber kann damit bei Aufzählbarkeit und Akzeptierbarkeit von Synonymen gesprochen werden.

Die Entscheidbarkeit ist anders, da für eine Sprache L die Turing-Maschine irgendwann mit einem bestimmten Ergebnis anhalten muss, wenn das Wort in L liegt und mit einem anderen Ergebnis anhalten muss, wenn das Wort nicht in L liegt. Das „und“ macht es anders, da so die Akzeptierbarkeit bzw. Aufzählbarkeit von L (wie vorher auch) und zusätzlich die Akzeptierbarkeit bzw. Aufzählbarkeit des Komplements von L gefordert wird.

Damit ist L entscheidbar, wenn L akzeptierbar und auch das Komplement von L akzeptierbar ist.

Frage: Ich finde das Thema mit der Entscheidbarkeit ziemlich schwer, wie sehen Sie das bei den anderen Themen.

Antwort: Vom Inhalt her haben Sie Recht, das ist das schwerste Kapitel der Veranstaltung. Die nachfolgenden sind meist deutlich leichter. Ich habe das Thema an den Anfang der Veranstaltung gestellt, da die Aussage „es gibt viele Eigenschaften von Programmen, die wir nicht automatisch nachweisen können“ die zentrale Aussage der Veranstaltung ist, die gerade auch in der Praxis im Hinterkopf bleiben muss. (Weiterhin sind Sie ja am Anfang des Semesters noch am Frischesten :))

Frage: Gibt es auch zu den anderen Aufgaben Beispiellösungen?

Antwort: Ist aktuell nicht geplant. Sie können aber Ihre Lösungen mit mir besprechen (VL-Zeit und Übungszeit). Das ist auch möglich, wenn Sie keinen Ansatz gefunden haben.

Frage: Gibt es für die Reduktionen ein Kochrezept?

Antwort: Ganz allgemein natürlich nicht. Trotzdem besteht immer die Idee mit einer Turing-Maschine ein Problem der einen Art in ein Problem der anderen Art zu verwandeln. Statt der formalen Forderung:

Definition: Gegeben seien Sprache1 und Sprache2 nicht notwendigerweise über den gleichen Alphabeten Alphabet1 und Alphabet2 sowie eine **total berechenbare Funktion**

$f: \text{Alphabet1}^* \rightarrow \text{Alphabet2}^*$; gelte dann für alle $w \in \text{Alphabet1}^*$
 $w \in \text{Sprache1}$ genau dann wenn $f(w) \in \text{Sprache2}$,

Kann man auch äquivalent versuchen zu zeigen:

$w \in \text{Sprache1}$ daraus folgt $f(w) \in \text{Sprache2}$ und

$w \notin \text{Sprache1}$ daraus folgt $f(w) \notin \text{Sprache2}$

Frage: Warum wird die Software als Prototyp bezeichnet?

Antwort: Ist genauer ein funktionaler Prototyp, der die (von mir gewünschte) Basisfunktionalität vollständig umsetzt, also komplett nutzbar ist, aber an vielen Stellen Optimierungen benötigt. Dies umfasst zumindest deutlich zu lange Methoden, eine zu geringe Testüberdeckung und die Performance diverser Methoden. Gerade bei der Performance wurde auf die Machbarkeit und eine erste einfache Lösung fokussiert; dies wäre aber im nächsten Schritt zu überarbeiten. Dies ist also kein Wegwerf-Prototyp der für Beispiel-Use-Cases die Machbarkeit zeigt, aber könnte zum wegwerfen animieren, um z. B. den domänengetriebenen Ansatz komplett umzusetzen. Da eine solche Überarbeitung Zeit kostet, ist unklar, wann sie in der Prioritätenliste weiter nach oben wandert.