Übungen zu Einführung in die Informatik I

Aufgabe 13 **Textersetzungssysteme**

Das hier zu entwickelnde Textersetzungssystem erwartet als Eingabe eine Strichzahl der Länge n , die in spitze Klammern eingeschlossen ist (also z.B. $\langle ||| \rangle$), als Repräsentation der natürlichen Zahl n . Gesucht ist ein Textersetzungssystem, das ermittelt, ob die Zahl n gerade oder ungerade ist. Die Ausgabe der Berechnung ist also entweder „gerade“ oder „ungerade“.

- Formulieren Sie die Grundidee des Algorithmus in Umgangssprache.
- Geben Sie den Zeichenvorrat und die Regeln zur Lösung dieses Problems an.
- Wenden Sie Ihr Semi-Thue-System auf das Wort $\langle ||||| \rangle$ an.

Aufgabe 14 **Markov**

Sei der Markov-Algorithmus M über den Zeichenvorrat $V = \{a, b\}$ und folgender Regelmenge T gegeben:

$$T = \left\{ \begin{array}{l} ba \rightarrow ab \\ aa \rightarrow \varepsilon \\ bb \rightarrow \varepsilon \\ a \rightarrow a \\ b \rightarrow b \end{array} \right\}.$$

- Wenden sie den Markov-Algorithmus M auf die Wörter ba , $abaaba$ und $baaab$ an.
- Geben Sie die Menge aller Wörter $S \subset V^*$ an, für welche M mit dem Ergebnis ε terminiert.
- Für welche Wörter terminiert M , wenn wir die Regel $aa \rightarrow \varepsilon$ entfernen?

Aufgabe 15 **Substitution mittels Markov-Algorithmen**

Markov-Algorithmen sind spezielle Textersetzungssysteme, die nach der Markov-Anwendungsstrategie arbeiten. Darüber hinaus sind bei Markov-Algorithmen spezielle Terminierungsregeln der Form $v \rightarrow .w$ erlaubt. Kommt in einer Berechnung eine solche Punktregel zur Anwendung, so terminiert die Berechnung unmittelbar danach. Wir betrachten in dieser Aufgabe eingeschränkte Boolesche Terme über den Identifikatoren $ID = \{x_1, \dots, x_n\}$ und mit leerer Menge von Bezeichnungen für atomare elementare Aussagen, d.h. $E = \{\}$.

- a) Geben Sie für einen vorgegebenen Booleschen Term t_1 und einen vorgegebenen Identifikator x_1 einen Markov-Algorithmus an, der angewendet auf beliebige Boolesche Terme t das Ergebnis der Substitution $t[t_1/x_1]$ als Ausgabe liefert.
- b) Wie läßt sich die Aufgabe (a) mit einem Textersetzungssystem ohne Verwendung einer Terminierungsregel lösen?
- c) Formulieren Sie einen Markov-Algorithmus, der für vorgegebene Terme $t_1, \dots, t_n$ und vorgegebene Identifikatoren $x_1, \dots, x_n$ angewendet auf beliebige Boolesche Terme t das Ergebnis der simultanen Substitution $t[t_1/x_1, \dots, t_n/x_n]$ als Ausgabe liefert.
- d) Wenden Sie den Algorithmus aus (c) an für das Beispiel $(x \wedge y)[y/x, x/y]$.

Aufgabe 16 (P) Markov-Algorithmus

- a) Geben Sie einen Markov-Algorithmus an, der für Wörter über dem Zeichenvorrat $\{a, b, c\}$ mit dem leeren Wort anhält, wenn das Wort gleichviele 'a' enthält, wie 'b' und 'c' zusammengenommen (es soll also gelten: $\text{Anzahl}(a) = \text{Anzahl}(b) + \text{Anzahl}(c)$). Andernfalls soll der Algorithmus mit dem Ergebnis mit einem von ε verschiedenen Wort terminieren.
- b) Im Verzeichnis `/usr/proj/infol/ws01/uebungsblatt04` steht Ihnen das Goferscript `markov.gs` zur Verfügung. Es enthält, neben verschiedenen Hilfsfunktionen die Funktion

```
markov :: String -> [(String, String, Bool)] -> (String, String),
```

die den Markov-Algorithmus implementiert.

Beschreibung der Funktion markov: Mit dem ersten Argument dieser Funktion wird das zu untersuchende Wort übergeben. Das zweite Argument ist eine Sequenz von 3-Tupel. Ein Element dieser Sequenz modelliert eine Regel des Markov-Algorithmus in folgender Weise: Das erste Element des Tupels enthält die linke Seite einer Regel und das zweite die rechte Seite. Das dritte Element ist vom Typ `Bool`; handelt es sich um eine haltende Regel nimmt dieses Element den Wert `True`, ansonsten den Wert `False` an. Das Ergebnistupel enthält in der ersten Komponente das Resultat des Algorithmus; in der zweiten Komponente wird mitgeteilt, ob der Algorithmus mit einer haltenden Produktion oder wegen Nichtanwendbarkeit terminiert hat.

Aufgabenstellung: Schreiben Sie ein Goferscript, in dem die Regeln aus Teilaufgabe a) in einer für die Funktion `markov` geeigneten Form definiert werden und testen Sie ihre Regeln an zwei Wörtern, wobei der Markov-Algorithmus in einem Fall mit ε , im anderen Fall mit einem von ε verschiedenen Wort terminieren soll.