## 1 Hashtabellen (5 Punkte)

Realisiert den ADT für das Lexikon (s. Übung 11) mit Hashtabellen. Zusätzlich zu der Funktion `find-in-lexikon` soll es nun auch `find-all-word-with-cat` geben (zweistellig: ein Lexikon und eine Kategorie), die eine Liste aller Wörter mit dieser Kategorie ausgibt. Dabei darf *nicht* das ganze Lexikon durchsucht werden! Hinweis: Beim Abspeichern der Worte müßt ihr die Worte auch nach Kategorien einsortieren, z.B. in eine zweite Hashtabelle.

## 2 Parsen mit Lexikon und Grammatik (5 + 10 Punkte)

Jetzt wo wir einen ADT zum Lexikon und zur Grammatik haben, können wir einen Parser verwenden, der diese ADTs anspricht. Musterlösungen zu den ADTs für Lexikon und Grammatik findet ihr auf der Webseite als `lexicon.lisp` und `grammar.lisp`. Dort steht auch ein passender Parser als `parser.lisp`. Versucht aber, eure eigenen ADTs zu verwenden.

Schreibt für Lexikon und Grammatik je eine Funktion, die einen Pfadnamen als Argument nimmt, die Datei einliest und ein Lexikon (bzw. Grammatik) zurückgibt. Beispieldateien (beachtet das Format der Dateien!!) stehen auf dem Web als `lexicon.txt` und `grammar.txt`.

Schreibt nun eine eigene Grammatik für eine natürliche Sprache eurer Wahl, die mit Fragesätzen und Aussagesätzen sowie Nomen, Artikeln, Adjektiven, Verben und Präpositionen umgehen kann (wenigstens 10 Regeln und ein Lexikon mit wenigstens 10 Worten) und verwendet sie mit dem vorgegebenen Parser (zeigt einen Lauf mit Traces).

## 3 Schmökaufgabe (3 Punkte)

Löst Aufgabe 1 mit `assoc`-Listen statt mit Hashtabellen.

Abgabetermin ist der 10. Februar 2000