

Künstliche Intelligenz

Blatt 1

Prof. Dr. M. Tüxen

In diesem Praktikum beschäftigen Sie sich mit der Bearbeitung von Listen in Lisp. Ferner machen Sie sich mit der Entwicklungsumgebung vertraut.

Listen können zur Repräsentation von endliche Mengen benutzt werden. Dabei kommt in der Liste kein Element mehrfach vor und die Reihenfolge der Elemente in der Liste ist irrelevant. Implementieren Sie mit Hilfe von `car`, `cdr`, `cons`, `null`, `equalp` und `if` die folgenden mengentheoretischen Operationen. Dabei repräsentiert `e` ein Element und `l`, `l1`, und `l2` jeweils Mengen.

Ferner lernen Sie den Umgang mit Arrays und Schleifen.

Testen Sie ihre Funktionen an einfachen Beispielen, die Sie als Kommentar zur Implementation hinzufügen. Bearbeiten Sie die Aufgaben in der angegebenen Reihenfolge. Dabei können Sie die Funktionen aus vorherigen Aufgaben benutzen.

1. Schreiben Sie ein Prädikat `my-elementp (e l)`, das `t` zurückgibt, falls `e` Element der Menge `l` ist. Es wird also berechnet, ob $e \in l$ gilt.
2. Schreiben Sie ein Prädikat `my-subsetp (l1 l2)`, das `t` zurückgibt, falls `l1` eine Teilmenge von `l2` ist. Es wird also berechnet, ob $l1 \subseteq l2$ gilt.
3. Schreiben Sie eine Funktion `my-adjoin (e l)`, die diejenige Menge zurückgibt, die durch Hinzufügen von `e` zur Menge `l` entsteht. Es wird also die Menge $\{e\} \cup l$ berechnet.
4. Schreiben Sie eine Funktion `my-union (l1 l2)`, die die Vereinigungsmenge von `l1` und `l2` zurückgibt. Es wird also die Menge $l1 \cup l2$ berechnet.
5. Schreiben Sie eine Funktion `my-intersection (l1 l2)`, die die Schnittmenge von `l1` und `l2` zurückgibt. Es wird also die Menge $l1 \cap l2$ berechnet.
6. Schreiben Sie eine Funktion `my-set-difference (l1 l2)`, die die Mengendifferenz von `l1` und `l2` zurückgibt. Es wird also die Menge $l1 \setminus l2$ berechnet.
7. Schreiben Sie eine Funktion `my-powerset (l)`, die die Potenzmenge von `l` zurückgibt. Es wird also die Menge $\mathcal{P}(l)$ berechnet. Hier kann die Benutzung einer Hilfsfunktion von Vorteil sein.

8. Implementieren Sie die Collatzfunktion `collatz (n)`, die für gerade natürliche Zahlen n die Zahl $n/2$ zurückgibt und für ungerade natürliche Zahlen n die Zahl $3 \cdot n + 1$ zurückgibt.
9. Schreiben Sie eine Funktion `iter (f x n)`, die für eine Funktion f , eine natürliche Zahl n und einen Startwert x die Liste $(x, f(x), f(f(x)), \dots, f^n(x))$ berechnet. Überprüfen die an Beispielen die Collatzvermutung.
10. Schreiben Sie eine Funktion `mat-add (m1 m2)`, die die Matrizenaddition implementiert. Die Funktion soll für beliebige Matrizen funktionieren, für die die Matrizenaddition definiert ist. Nutzen Sie `assert` um dies zu überprüfen.
11. Schreiben Sie eine Funktion `mat-mul (m1 m2)`, die die Matrizenmultiplikation implementiert. Die Funktion soll für beliebige Matrizen funktionieren, für die die Matrixmultiplikation definiert ist. Nutzen Sie `assert` um dies zu überprüfen.
12. Schreiben Sie eine Funktion `mat-pow (m n)`, die die n -te Potenz einer quadratischen Matrix berechnet. Dabei ist n eine natürliche Zahl. Minimieren Sie die Anzahl der notwendigen Matrixmultiplikationen.