

Gesamtpunktzahl: 30

Abgabe der Lösungen bis zum 14.12.2015

**Aufgabe 2:** Anwendung (1): Münzautomat

14 Punkte

maximale Bearbeitungszeit: 80 Minuten

Beim Einkauf einer Ware gibt es meist verschiedene Möglichkeiten zur Auszahlung des Wechselgeldes (Differenz zwischen dem vom Käufer gegebenen Geldbetrag und dem Preis der Ware).

1. Implementieren Sie ein Prädikat, das für ein gegebenes Münzsystem (z.B. die Euro-Münzen) alle möglichen Zerlegungen des Wechselgeldes als Liste von Münzwerten berechnet.

**Hinweis:** Repräsentieren Sie das Münzsystem als sortierte Liste von Münzwerten und verwenden Sie das Prädikat `member/2` als Generator, um für die verschiedenen Münzwerte zu testen, ob sie als Wechselgeld infrage kommen.

2. Stellen Sie in Ihrer Implementation sicher, dass Ihr Prädikat zwar alle möglichen Zerlegungen des Betrags in Münzen ermittelt, aber keine Mehrfachresultate, d.h. auch keine Permutationen der Ergebnisliste ausgibt.

Begründen Sie Ihre Entscheidung für den von Ihnen verwendeten Lösungsansatz.

**Hinweis:** Permutationen lassen sich z.B. dadurch vermeiden, dass nur geordnete Listen von Wechselgeldmünzen mit monoton fallenden oder wachsenden Münzwerten betrachtet werden. Dann lassen sich z.B. durch eine zusätzliche Bedingung alle Listen blockieren, die von der vereinbarten Sortierreihenfolge abweichen. Alternativ dazu kann man aber auch der nachfolgenden (rekursiven) Abarbeitung diejenigen Münzen mitteilen, die aufgrund der bereits ausgewählten Münzen noch zulässig sind.

3. Modifizieren Sie Ihre Prädikatsdefinition so, dass für das System der Euro-Münzen immer nur die optimale Zerlegung für das Wechselgeld ermittelt wird, d.h. diejenige, die mit der kleinsten Zahl von Münzen auskommt.

4. In der Hamburger Partyszene ist das Gerücht aufgetaucht, dass es angeblich Bestrebungen gibt, eine 40-Cent-Münze einzuführen. Diskutieren Sie, ob Ihr Prädikat aus Teilaufgabe 3 auch für das so erweiterte Münzsystem immer noch die optimale Zerlegung findet. Welche Bedingung muss das Münzsystem erfüllen, damit Sie die Optimalität des Ergebnisses garantieren können? Was könnten Sie tun, um die Optimalität auch für beliebige Münzsysteme zu garantieren?
5. Erweitern Sie Ihre Prädikatsdefinition so, dass sie in einem Automaten eingesetzt werden kann und bei der Herausgabe des Wechselgeldes berücksichtigt, ob die jeweils benötigten Münzen im Geldspeicher noch vorhanden sind. Ihr Prädikat sollte auch berechnen, wieviele Münzen der verschiedenen Sorten der Münzspeicher nach Herausgabe des Geldes noch enthält.

**Hinweis:** Modellieren Sie den Geldspeicher durch eine Liste, die Paare aus dem Münzwert und der Anzahl der noch vorhandenen Münzen dieses Werts enthält.

Bonus: Erweitern Sie Ihr Prädikat aus der vorangegangenen Teilaufgabe zu einem kompletten Warenautomaten, der für einen zu bezahlenden Preis eine Liste von Münzen entgegennimmt und eine Liste der herauszugebenden Münzen berechnet. Wenn nicht ausreichend viele Münzen eingeworfen wurden, soll das Prädikat scheitern. (5 Punkte)

## Aufgabe 2: Anwendung (2): Buchstabenbäume

16 Punkte

maximale Bearbeitungszeit: 100 Minuten

Buchstabenbäume (Tries) können für einen effizienten Zugriff zu Datenbankinformationen genutzt werden, falls die Zugriffsschlüssel der Datenbank sehr oft identische Anfangssequenzen besitzen. Dies ist z.B. bei den Wortformen in vielen (flek-tierenden) Sprachen der Fall:

*dein, deine, deinem, deinen, deiner, deines, ....*

Ein Beispiel für einen Buchstabenbaum auf der Basis von Listen finden Sie auf S. 124 im Skript.

Implementieren Sie eine Prädikatsammlung zum Aufbau eines solchen Buchstabenbaums ausgehend von einer Sammlung von Schlüssel-Wert-Paaren. Die Datei `woerterbuch.pl` enthält ein (vereinfachtes) Aussprachewörterbuch für deutsche Wortformen, wie es in vielen Anwendungen der Sprachtechnologie benötigt wird:

```
...
entry(dein, [d, aI, n]).
entry(deine, [d, aI, n, @]).
```

```

entry(deinem, [d, aI, n, @, m]).
entry(deinen, [d, aI, n, @, n]).
entry(deiner, [d, aI, n, '6']).
entry(deines, [d, aI, n, @, s]).
...

```

Die Fakten dieser Datenbank definieren eine Zuordnung zwischen der orthographischen Form eines Wortes (z.B. **deinem**) und seiner Aussprache (gegeben als Liste von Lautsymbolen, z.B. **[d, aI, n, @, m]**). Im Gegensatz zur Realität in den natürlichen Sprachen wird hier angenommen, dass diese Zuordnung eineindeutig ist.

Mit Hilfe des ebenfalls in der Datei **woerterbuch.pl** gegebenen Prädikats **dictionary/1** können Sie eine solche Faktensammlung auch in eine Liste von Schlüssel-Wert-Paaren umwandeln:

```

[entry(dein, [d, aI, n]),
 entry(deine, [d, aI, n, @]),
 entry(deinem, [d, aI, n, @, m]),
 entry(deinen, [d, aI, n, @, n]),
 entry(deiner, [d, aI, n, '6']),
 entry(deines, [d, aI, n, @, s]) ]

```

Die Art der mit einem Schlüssel verbundenen Nutzinformation ist hierbei durch die konkrete Anwendung vorgegeben, soll im Prinzip aber beliebig sein.

1. Entscheiden Sie sich, welche der beiden Argumentpositionen in der Struktur **entry/2** als Schlüssel und welche als Wert verwendet werden sollen. Grundsätzlich ist diese Entscheidung beliebig, allerdings werden orthographische Schlüssel eher in der Sprachsynthese, lautsprachliche eher in der Spracherkennung benötigt.
2. Analysieren Sie die Struktur eines Buchstabenbaums und beschreiben Sie diese durch eine kontextfreie Grammatik oder einen Typtest. Überlegen Sie dabei insbesondere, wie sie die jeweilige Nutzinformation geeignet in den Baum integrieren können.
3. Implementieren Sie ein Prädikat, das für einen als Liste von Einzelzeichen gegebenen Schlüssel und eine mit diesem Schlüssel verbundene Nutzinformation einen (elementaren) Buchstabenbaum konstruiert, z.B.

```

? word2trie([b,a,u,e],NUTZINFORMATION,Trie).
  Trie = [[b,[a,[u,[e,[*,NUTZINFORMATION]]]]]] ;
  false.

```

4. Implementieren Sie ein vierstelliges Prädikat, das ein neues Schlüssel-Wert-Paar in einen gegebenen Buchstabenbaum einträgt und auf diese Weise einen neuen (d.h. ergänzten) Buchstabenbaum berechnet:

`insert_entry(+Schluessel,+Nutzinformation,+Alter_Trie,?Neuer_Trie)`

Arbeiten Sie dabei den als Liste von Einzelzeichen gegebenen Schlüssel rekursiv ab und unterscheiden Sie für jedes Einzelzeichen zwischen drei Fällen:

- a) Das aktuelle Zeichen ist bereits im alten Buchstabenbaum enthalten. Wählen Sie den zugehörigen Teilbaum und tragen Sie ihn in aktualisierter Form in den neu aufzubauenden Buchstabenbaum ein.

Hinweis: Zum Überprüfen, ob für einen gegebenen Zeichen bereits ein passender Teilbaum existiert, können Sie das eingebaute Prädikat `select/3` verwenden. Es entfernt den gesuchten Teilbaum aus der Liste der im Baum enthaltenen Teilbäume. Der aktualisierte Teilbaum kann dann nach dem Hinzufügen des neuen Eintrags wieder in den Gesamtbaum integriert werden.

- b) Das aktuelle Zeichen ist noch nicht im Buchstabenbaum enthalten. Dann muss der aktuelle Teilbaum um einen neuen Ast für den noch einzutragenden Rest des Schlüssels ergänzt werden (vgl. hierzu Teilaufgabe 3).
- c) Der einzutragende Schlüssel ist komplett abgearbeitet, d.h. er ist bereits im alten Buchstabenbaum vollständig enthalten. Dann muss nur noch der aktuelle Teilbaum um die Nutzinformation und ggf. um eine geeignete Endemarkierung ergänzt werden.

Dokumentieren Sie ausführlich die Vorgehensweise des von Ihnen implementierten Algorithmus.

5. Implementieren Sie ein Prädikat, das die gegebenen Schlüssel-Wert-Paare in einen Buchstabenbaum einträgt.

Hinweis: Falls Ihr Schlüssel ein Atom ist, können Sie dieses mit Hilfe des eingebauten Prädikats `atom_chars/2` in eine Liste von Einzelzeichen umwandeln. Falls Sie die in einer Liste enthalten Atome zu einem einzelnen Atom zusammenfügen wollen, können Sie dafür das eingebaute Prädikat `atomic_list_concat/2` verwenden.

6. Modifizieren Sie das im Skript angegeben Prädikat `word/2` so, dass Sie damit auch die Nutzinformation aus einem Buchstabenbaum auslesen können.

Bonus: 1 Überlegen Sie sich eine Möglichkeit, wie unterschiedliche Nutzinformationen zum gleichen Schlüssel in einem Buchstabenbaum verwalten kann. (1 Punkt)

- Bonus: 2 Berechnen Sie für die in der Datei `woerterbuch.pl` gegebene Liste von Schlüssel-Wert-Paaren einen Binärbaum und einen Buchstabenbaum. Ermitteln Sie den Zeitaufwand für den Aufbau der beiden Indexstrukturen mit dem gegebenen Wörterbuch. (3 Punkte)
- Bonus: 3 Untersuchen Sie für eine hinreichend große Stichprobe, welche der beiden Indexstrukturen einen effizienteren Zugriff zu den Indexinformationen ermöglicht. (3 Punkte)
- Bonus 4: Um einen modifizierten Teilbaum nichtdestruktiv in den restlichen Baumkontext integrieren zu können, muss die aktuelle Liste der alternativen Teilbäume kopiert werden. Nur so kann erreicht werden, dass der ursprüngliche Baum nicht zerstört wird. Dies ist insbesondere in nebenläufigen Umgebungen wichtig, um eine nebeneffektfreie Verarbeitung zu garantieren, ohne dafür Sperrmechanismen einsetzen zu müssen. Analysieren Sie beispielsweise die Vorgehensweise des Prädikats `select/3` und überzeugen Sie sich davon, dass hier wirklich eine Kopie der Liste (ohne das jeweils ausgewählte Element) angelegt wird.
- Überlegen Sie, durch welche Maßnahmen dieser Kopieraufwand minimiert werden kann. (4 Punkte)