

Team: 11, Mesut und Anton

Team: 11, Mesut Koc und Anton Kirakozov

Aufgabenaufteilung: Alle Aufgaben gemeinsam bearbeitet (*Aufgabe_2.pl und Dokumentation*)

Quellenangaben: <http://www.cis.upenn.edu/~matuszek/Concise%20Guides/Concise%20Prolog.html>

Bearbeitungszeitraum: Alle Aufgaben wurden gemeinsam bearbeitet (*siehe gemeinsame Bearbeitungszeit*)

Gemeinsame Bearbeitungszeit: Vom 20.04.15 bis 27.04.15 ca. 16h & 28.04.15 ca. 2h

Aktueller Stand: Alle Funktionen sind implementiert, soweit getestet und funktionsfähig. Die Dokumentation ist vollständig beschrieben.

Konzepte unserer fertigen Aufgaben:

Aufgabe 1 Teil 1

2.1.1) Die Aufgabe erfordert ein Prädikat, das Vornamen, Namen, Geburts- und Todesjahr aller Komponisten auf dem Bildschirm ausgibt. Dabei sind die Komponisten in einer strukturierten Datenbank (*cd.pl*) gespeichert. Das Schema ist eigentlich recht einfach zu verstehen: Jeder Komponist hat einen identifizierbaren Schlüssel und Informationen wie Namen usw. und diese sollen ausgegeben werden. Wir wollen alle Datensätze, wo KNR nur einmal vorkommt. Da die KNR auch „Null“ sein kann, haben wir drauf Rücksicht genommen. Somit stellen wir sicher, dass jeder Komponist einmal ausgegeben wird.

2.1.2) Hier haben wir die Anforderungen, dass alle Komponisten ausgegeben werden müssen, die im 18. Jahrhundert geboren sind. Aus dem Text heraus konnten wir schon einige Merkmale herausziehen (z.B. *geboren >=1700 & gestorben <= 1799*). Deswegen haben wir vorerst unser Pseudocode aufgestellt:

*wenn Komponist genau 1700 oder > geboren UND
gestorben genau am 1799 oder < dann gebe alle Komponisten zu dem Zweig aus*

Wir vergleichen das Geburts- und Todesjahr aller Komponisten. Dementsprechend geben wir alle Komponisten aus, die zu dem Vergleich ein Ergebnis liefern.

2.1.3) In dieser Teilaufgabe sollen alle Titel der CD's ausgegeben werden, die von bestimmten Firmen produziert werden. Somit kann jede Firma abgefragt werden und man erhält die genauen Titel der CD's. Diese Aufgabe unterscheidet sich im wesentlichen von der Teilaufgabe 2.1.2, da sich die Abfragen anders gestalten.

2.1.4) Bei dem Prädikat **print_soli(Titel)** sollen alle Solisten ausgegeben werden, die in einem bestimmten Stück auftreten. Bei dieser Art von Abfrage müssen wir einige Faktoren beachten. Wir betrachten mehrere Entitätstypen gleichzeitig und vergleichen deren Primärschlüssel, damit wir eine korrekte Abfrage gewährleisten können. Falls bei der Abfrage kein Ergebnis geliefert wird, liegt es daran, dass bestimmte Solisten in den abgefragten Stücken und Aufnahmen nicht vertreten sind.

gehe jeden Datensatz im Stück durch den ersetze den Wert für „Hersteller“ durch HListe durch.

2.1.5) Bei dem Prädikat **print_komp(Name)** geben wir alle Komponisten auf dem Bildschirm aus, bei deren Werk ein bestimmter Solist (*Name*) mitgewirkt hat. Wir gehen alle Datensätze von „Solist“ durch, zudem der (*Name*) passt und vergleichen dementsprechend den Fremdschlüssel KNR!komponist mit dem Primärschlüssel KNR aus Komponist. Die eindeutig identifizierbaren Datensätze geben wir aus bzw. nur den Namen.

*gehe alle Datensätze durch von Solist zu dem der Name passt
ist der Fremdschlüssel KNR == Primärschlüssel KNR?
gebe den Namen zu den identifizierbaren Daten aus*

Aufgabe 2 Teil 2

2.2.1-2.2.2) In diesem Teil der Aufgabe geht es darum aussagenlogische Prädikate zu entwerfen. Wir fassen die Teilaufgabe 2.2.1 und 2.2.2 zusammen, da wir außer für die Prädikate *mnot*, *and* und *or* keine weiteren logischen Operatoren mehr verwenden dürfen. Des Weiteren soll *mnot* unsere Negationsfunktion darstellen, die wir selber entwickeln, weil Prolog **NOT** kennt und wir das nicht verwenden sollen. Für die Prädikate *nand*, *nor*, *xor*, *impl.* und *aequiv.* verwenden wir ausschließlich die Prädikate *mnot*, *and* und *or*. Anhand der Beispiele können wir darstellen, wie unsere aussagenlogischen Prädikate funktionieren.

mnot : False → True
and: False, False → False
or: True, False → True

nor: False, True → True
xor: False, False → False
implikation: True, False → False

2.2.3) Das Prädikat **tafel/3** soll dazu dienen unsere logischen Ausdrücke besser visuell darstellen zu können. Davor hatten wir die Möglichkeit einen bestimmten Zustand zu erzeugen und Prolog hat uns diesen Zustand ausgewertet und dazu das Ergebnis geliefert. Jetzt können wir komplett alle Zustände graphisch darstellen, die bei zwei aussagenlogischen Variablen möglich sind. Verwendet man zwei Variablen wären es somit vier Kombinationen an Wahrheitswerten, die zustande kommen können. Des Weiteren können wir uns bei der Tafel entscheiden, welche logische Operation durchgeführt werden soll (*und, oder usw.*).

2.2.4) Das Prädikat **tafel/4** gestaltet sich genauso, wie **tafel/3** mit dem Unterschied, dass es anstatt zwei Variablen drei sein müssen. Die Auswertung und die Operatoren bleiben wie in der Aufgabe davor gleich.

2.2.5) Bei dem letzten Prädikat **tafel/2** soll die Darstellung für n-Variablen gestaltet werden. Wir interpretieren diese Aufgabe so, dass wir uns mehrere Klauseln mit zwei logischen Operatoren vorstellen können, dass würde wie folgt aussehen: $(A \vee B) \wedge (C \vee D)$ usw. Da es mehr als drei Variablen geben kann, entwickeln wir das Prädikat für N-Variablen. Die Auswertung und auch die Operatoren bleiben, wie auch in den vorherigen Aufgaben gleich. Anhand eines Beispiels können wir unsere Annahme mehr verdeutlichen:

And(true or false),(true or false) -> true