Programmentwurf Evidenzen

# Benutzung des Programms

Die Buchempfehlung ist ein Java-Programm, gepackt als JAR inklusive allen Bibliotheken. Um Empfehlungen ausgeben zu können, muss zuerst die Datenbank mit den Beispieldaten gefüllt werden. Dazu führt man das Programm mit den Argumenten create\_db <filename> aus. Dieser Schritt ist nur einmalig nötig, die Datenbank bleibt über mehrere Sitzungen gespeichert.



Um eine Empfehlung auszusprechen, wird eine CSV-Datei ähnlich der Beispieldaten als Input benötigt. Diese hat die Spalten Altersgruppe, Geschlecht, Verheiratet, Kinderzahl, Abschluss, Beruf, Familieneinkommen. Die Spalten können leer sein, die Strichpunkte zum Trennen der Spalten müssen jedoch vorhanden sein. Die Argumente zum Aussprechen von Empfehlungen sind recommend <filename>.



Die Details zu den Empfehlungen werden in die Konsole geschrieben. Zusätzlich wird eine Datei neben der Eingabedatei angelegt. Diese hat denselben Dateinamen wie die Eingabedatei, dem allerdings die Dateiendung .out hinzugefügt wird. In unserem Beispiel mit der Eingabedatei testinput.csv heißt die Ausgabedatei testinput.out.csv. Die Ausgabedatei enthält dieselben Daten wie die Eingabedatei, ergänzt mit der Spalte Buchempfehlung.

# Implementierung von Mengen

Wir haben den Ansatz gewählt, die Menge an Alternativen und die gültigen Werte in der CSV-Datei mittels Java-Enums abzubilden, da es sich bei diesen Werten um einen kleinen Bereich gültiger Werte handelt. Dafür wurde für jede Spalte ein Enum erstellt, das alle möglichen Werte enthält. Das Enum hat eine Methode getEnumType(), die statisch den Namen der Spalte in der Datenbank zurückgibt.

Beim Einlesen aus der CSV-Datei müssen die Strings in Enums umgewandelt werden. Da die Strings teilweise mit Zahlen beginnen und Leerzeichen enthalten, dies für die Werte der Enums allerdings nicht zulässig ist, muss jedem Wert eine String zugeordnet werden. Dies wird über den Konstruktor des Enums gemacht, dieser speichert den Wert in einer lokalen Variablen. Die Methode fromString() kann nun einen String entgegennehmen und den Wert des Enums zurückgeben, dessen lokale Variable mit dem Parameter übereinstimmt. Unglücklicherweise lässt Java es nicht zu, dass Enums voneinander erben, deswegen muss dieser Code für jedes Enum dupliziert werden.

Die Teilmengen von Omega werden mittels Java-Sets modelliert. Sets entsprechen den mathematischen Mengen und werden aus den Werten der Enums gebildet. Omega selbst entspricht dem Set aus allen Werten des Enums, die mit der vom Compiler generierten Methode values() abgerufen werden können.

# Basismaß und Dempster-Shafer