Introduction to
Data and Knowledge Engineering

TECHNISCHE UNIVERSITÄT DARMSTADT

Tutorium 7: Datalog

Aufgabe 1 Produktbewertungen

Um fingierte Kundenrezensionen bei Amazon.com zu erkennen, wollen wir die Reviews analysieren und bestimmte Kriterien verwenden. Gegeben sei eine Datenbasis, die Fakten in der folgenden Form enthält.

```
review( Review_Id, Reviewer_Id, Rating, Product_Id, Review_Body, Date)

product( Product_ID, Manufacturer, AVG_Rating, SalesRanking)

reviewer( Reviewer_ID, Total_Reviews)
```

Dabei stehen die Variablen für Konstanten mit den folgenden Bedeutungen.

- `Reviewer_ID`: Eindeutige Nummer für jeden Kunden bzw. Reviewer
- `Review_ID`: Eindeutige Nummer für jedes Review
- `Rating`: Eine ganze Zahl zwischen 1 und 5 (5: excellent, 4: good, 3: average, 2: bad, 1: useless)
- `Product_Id`: Eindeutige Nummer für jedes Produkt
- `Review_Body`: Inhalt des Reviews als String
- `Date`: Abgabedatum des Reviews
- `Manufacturer`: Herstellername
- `AVG_Rating`: Durchscnittliches Rating für das Produkt
- `SalesRanking` Rangzahlen zwischen 1 und 10.000. Je höher die Anzahl der verkauften Exemplare, desto kleiner die Rangzahl
- `Total_Reviews`: Anzahl von Reviews von einem Reviewer

Definieren Sie Prädikate, die die folgenden Anfragen beantworten.

a) `review_biased_reviewerA(Review)`: Finden Sie die Reviews, die von den Reviewern geschrieben wurden, die mindestens 1000 Reviews geschrieben haben.

b) `review_biased_reviewerB(Review)`: Finden Sie die Reviews, die von den Reviewern geschrieben wurden, die ein Produkt mehr als einmal und zwar am selben Tag bewertet haben.

c) `duplicates(Review)`: Finden Sie die Duplikate, d.h. Reviews mit demselben Inhalten.

d) `lowSalesRanking(Product)`: Finden Sie die Produkte mit niedriger Anzahl verkaufter Exemplare. Wir definieren dies als ein SalesRanking größer als 5.000.

e) `highRating_biased(Review)`: Finden Sie die Reviews mit einem hohen Rating (5) für Produkte mit lowSalesRanking.

f) `spam(Review)`: Finden Sie die Reviews, die mindestens eine der oben genannten Eigenschaften (a,b,c,e) besitzen.

g) Finden Sie Produkte von Apple und Sony, für die es Reviews gibt, die mit anderen Reviews im Inhalt genau übereinstimmen.

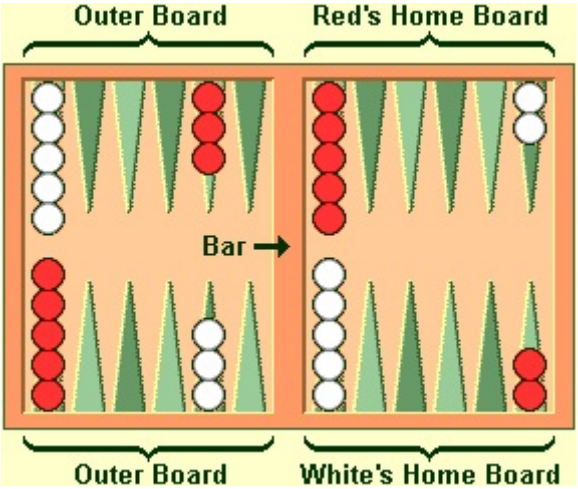
Hinweis: Wir können Reviews, Reviewer und Produkte durch eindeutige Nummern (IDs) identifizieren, z.B wenn wir nach Reviews mit bestimmten Eigenschaften suchen. Das Ergebnis ist eine Liste von Review-IDs.

Aufgabe 2 Negation

Backgammon ist ein Brettspiel, dessen direkte Vorläufer schon seit vielen Hunderten von Jahren gespielt werden. Das Spielbrett, siehe Abbildung unten, besteht aus 24 Feldern ("Zungen") auf denen bis zu fünf Spielsteine liegen können. Im Verlauf des Spiels können sich Steine außerdem auf der Bar befinden oder, gegen Ende, ganz vom Spielfeld verschwinden.

Die jeweils 15 Steine der beiden Speieler werden zu Beginn wie unten abgebildet auf die Felder verteilt. Bewegung der Spielsteine bestimmen die Spieler mit zwei Würfeln. Pro Würfel kann ein Spielstein um die Augenzahl weiter gerückt werden, wobei der untere Spieler (rot) im Uhrzeigersinn und der obere Spieler (weiss) gegen den Uhrzeigersinn zieht. Dabei kann man nur auf Felder ziehen, auf denen nicht mehr als ein gegnerischer Stein liegt. Liegt dort ein gegenerischer Stein, wird dieser geschlagen und auf die Bar gesetzt. Der Gegner darf erst dann wieder andere Steine ziehen, wenn er alle seine Steine von der Bar zurück ins Spiel gebracht hat, in dem er sie auf die für ihn ersten Felder zieht.

Sind alle Steine eines Spieler in seinen Home Board angekommen, kann er beginnen, sie aus dem Spielfeld herauszuziehen. Wer als erster keine Steine mehr auf dem Brett hat, hat das Spiel gewonnen.



Nehmen Sie an, in einer Datalog-Datenbank werden die Prädikate `player/2`, `color/2`, `point/2` und `home/2` verwendet. `player(P,C)` bedeutet, dass Spielsteine des Spielers `P` die Farbe `C` haben. `color(S,C)` bedeutet, dass Spielstein `S` die Farbe `C` hat. `point(S,P)` heißt, dass Spielstein `S` auf dem `P`-Point liegt, wobei die Bar auch als ein Point betrachtet wird. `home(S)` bedeutet, dass Spielstein `S` im Heimfeld liegt.

Definieren Sie folgende Prädikate.

a) `player_stone(P,S)` ist wahr, genau dann wenn Spielstein `S` dem Spieler `P` gehört.

b) `can_be_beaten(S)` ist wahr, genau dann wenn Spielstein S geschlagen werden kann. Das bedeutet, dass Spielstein S allein auf einem Point liegt.

c) `start_move_out(P)` ist wahr, wenn Spieler `P` seine eigenen Spielsteine hinauswürfeln kann, das heißt, wenn sich alle seine eigenen Spielsteine im Heimfeld befinden.

d) `player_win(P)` ist wahr, genau dann wenn Spieler `P` gewonnen hat, das heißt, es gibt keine Spielsteine mehr am Brett, die dem Spieler `P` gehören.

e) (**Extraaufgabe**) `can_move(A,P)` ist wahr, genau dann wenn Spieler A seine eigenen Spielsteine auf das Feld (Point) P ziehen kann. Das heißt, dass Feld P nicht von mehr als einem gegnerischen Stein besetzt ist.

Hinweise

Sie können alle hier aufgeführten Aufgaben mit dem Programm *Datalog Educational System* (`https://des.sf.net`), welches für verschiedene Betriebssystem zur Verfügung steht, ausprobieren. Bei der Verwendung ist zu beachten, dass Sie die Endung der Dateien richtig setzen (`.dl`) und dass das Programm sehr sensibel auf falsch gesetzte Leerzeichen reagiert. Zwischen Prädikatname und öffnende Klammer sollte z.B. kein Leerzeichen stehen.