

Gesamtpunktzahl: 20

Abgabe der Lösungen bis zum 11.11.2019

Aufgabe 1: Terminologische Grundlagen

2 Punkte

maximale Bearbeitungszeit: 10 Minuten

Erklären Sie die Begriffe "Suche", "Variable", und "Instanziierung" im Hinblick auf ihre Rolle in der Logikprogrammierung.

Aufgabe 2: Relationale Datenbanken: Sichten

6 Punkte

maximale Bearbeitungszeit: 30 Minuten

Die Datei `dateiverzeichnis.pl` enthält eine relationale Repräsentation für ein Dateiverzeichnis als Faktensammlung.

Definieren Sie für diese Datenbank die folgenden Prädikate mit Hilfe von Regeln. Geben Sie die möglichen Instanziierungsvarianten für Ihre Prädikatsdefinitionen an.

1. Ein Prädikat, das Dateinamen und Dateischlüssel ineinander umrechnen kann. Achten Sie dabei darauf, dass Dateinamen in einem Verzeichnisbaum nicht eindeutig sein müssen. In solchen Fällen, soll das Prädikat alternative Variablenbindungen für den Dateischlüssel erzeugen.
2. Ein Prädikat, das Verzeichnisnamen und Verzeichnisschlüssel ineinander umrechnen kann. Achten Sie darauf, dass auch Verzeichnisnamen in einem Verzeichnisbaum nicht eindeutig sein müssen.
3. Ein Prädikat, das für einen gegebenen Dateinamen, den Namen und den Schlüssel des zugehörigen Verzeichnisses ermittelt.
4. Ein Prädikat, das für einen gegebenen Verzeichnisnamen, den Namen und den Schlüssel des unmittelbar übergeordneten Verzeichnisses ermittelt.

Testen Sie Ihre Definitionen an geeigneten Beispielen. Erweitern Sie dazu bei Bedarf die Datenbasis so, dass alle wesentlichen Fälle berücksichtigt sind. Beobachten Sie im Trace, wie die Abarbeitung Ihrer Testanfragen durch das Prolog-System erfolgt.

Aufgabe 3: Aggregation in relationalen Datenbanken

4 Punkte

maximale Bearbeitungszeit: 30 Minuten

Definieren Sie für die Arbeit mit der Verzeichnis-Datenbank aus Aufgabe 2 die folgenden Prädikate. Geben Sie auch hierfür immer die möglichen Instanziierungsvarianten an.

1. Ein Prädikat, das eine Liste aller Dateinamen ermittelt, die in einem bestimmten Verzeichnis enthalten sind. Das Verzeichnis soll durch seinen Schlüssel identifiziert werden.
2. Ein Prädikat, das eine Liste der Namen aller Unterverzeichnisse ermittelt, die in einem bestimmten Verzeichnis enthalten sind. Das Verzeichnis soll wiederum durch seinen Schlüssel identifiziert werden.
3. Ein Prädikat, das die Anzahl aller Dateien in einem bestimmten Verzeichnis ermittelt. Das Verzeichnis soll auch hier durch seinen Schlüssel identifiziert werden.

Hinweis: Die Länge einer Liste können Sie mit dem vordefinierten Prädikat `length/2` ermitteln.

Aufgabe 4: Deduktive Datenbanken (1)

6 Punkte

maximale Bearbeitungszeit: 30 Minuten

Definieren Sie für die Datenbank `familie.pl` aus Aufgabenblatt 2 die folgenden Prädikate

1. `vorfahre_von(Vorfahre,Nachkommende)`, das für eine gegebene Person alle ihre Vorfahren berechnet.
Ist Ihre Implementierung richtungsunabhängig und terminierungssicher? Geben sie entsprechende Zusicherungen an.
2. `juengster_gemeinsamer_vorfahre(Person1,Person2,Juengster_Vorfahre)`, das den jüngsten gemeinsamen Vorfahren von zwei Personen berechnet.
3. `verwandt_mit(Person,Verwandter)`, wobei zwei Personen verwandt miteinander sind, wenn sie gemeinsame Vorfahren haben
4. Für den Fall, dass Ihre Implementation Doppelergebnisse produziert, Was ist die Ursache dafür? Wie lässt sich das Problem beheben?

Aufgabe 5: Relationale Datenbanken: Transaktionen

2 Punkte

maximale Bearbeitungszeit: 10 Minuten

Definieren Sie für die Arbeit mit einem Dateiverzeichnis ein Prädikat, das das Änderungsdatum eines Verzeichnisses auf das aktuelle Datum setzt. Achten Sie auf eine geeignete Fehlerbehandlung, falls das zu ändernde Verzeichnis nicht existiert.

Hinweis 1: Definieren Sie zur Fehlerbehandlung zuerst die Klausel, die das Verhalten des Prädikats im Normalfall beschreibt. Erweitern Sie diese Definition anschließend um zusätzliche Klauseln bzw. Teilziele mit den entsprechenden Bedingungen für die jeweiligen Fehlerfälle bzw. den Normalfall. Auf einer zusätzlich hinzugefügten Argumentstelle können Sie dann einen geeigneten Fehlercode, z.B.

- `transaktion_erfolgreich` bzw.
- `verzeichnis_existiert_nicht`

zurückgeben.

Hinweis 2: Nicht mehr benötigte Klauseln können Sie bei Bedarf mit Hilfe der Prädikate `retract/1` bzw. `retractall/1` aus der Datenbank entfernen.

Hinweis 3: Das aktuelle Datum kann über das Prädikat `date/1` abgefragt werden.

Zusatzaufgaben: nur für Interessenten

6 Punkte

maximale Bearbeitungszeit: 40 Minuten

Ergänzen Sie die Implementierung Ihres Dateisystems um weitere Funktionalitäten:

1. Ein Prädikat, das in einem gegebenen Verzeichnis ein neues Unterverzeichnis hinzufügt. Stellen Sie sicher, dass das übergeordnete Verzeichnis existiert, keine Doppeleintragungen in einem Verzeichnis vorgenommen werden und die Angaben für das Modifikationsdatum am Ende der Transaktion für alle beteiligten Verzeichnisse auf dem aktuellen Stand sind.

Hinweis: Mit dem Prädikat `flag/3` können Sie sich neue Verzeichnisschlüssel generieren:

```
?- flag(directories,_,30).      % Initialisieren des Zählers
true.
?- flag(directories,X,X+1).    % Abfragen und Erhöhen des Zählers
X = 30.
?- flag(directories,X,X+1).    % bei jedem Aufruf
X = 31.
?- flag(directories,X,X+1).    % . . .
X = 32.
...
```

2. Ein Prädikat, das in einem gegebenen Verzeichnis eine neue Datei hinzufügt. Stellen Sie auch hier sicher, dass das angegebene Verzeichnis existiert, keine Doppeleintragungen in einem Verzeichnis vorgenommen werden und alle Angaben zum Modifikationsdatum aktualisiert wurden.
3. Ein Prädikat, das den Verzeichniseintrag aktualisiert, wenn eine über ihren Schlüssel identifizierte Datei durch eine Applikation verändert wird.

4. Vergleichen Sie Ihre Lösungen zu den Teilaufgaben 1 bis 3 mit dem im Bereich der Datenbanken benutzten Begriff der Transaktion. Diskutieren Sie die Anforderungen an eine Transaktion, die von Ihrer Implementation nicht erfüllt werden.