Team: 11, Mesut und Anton

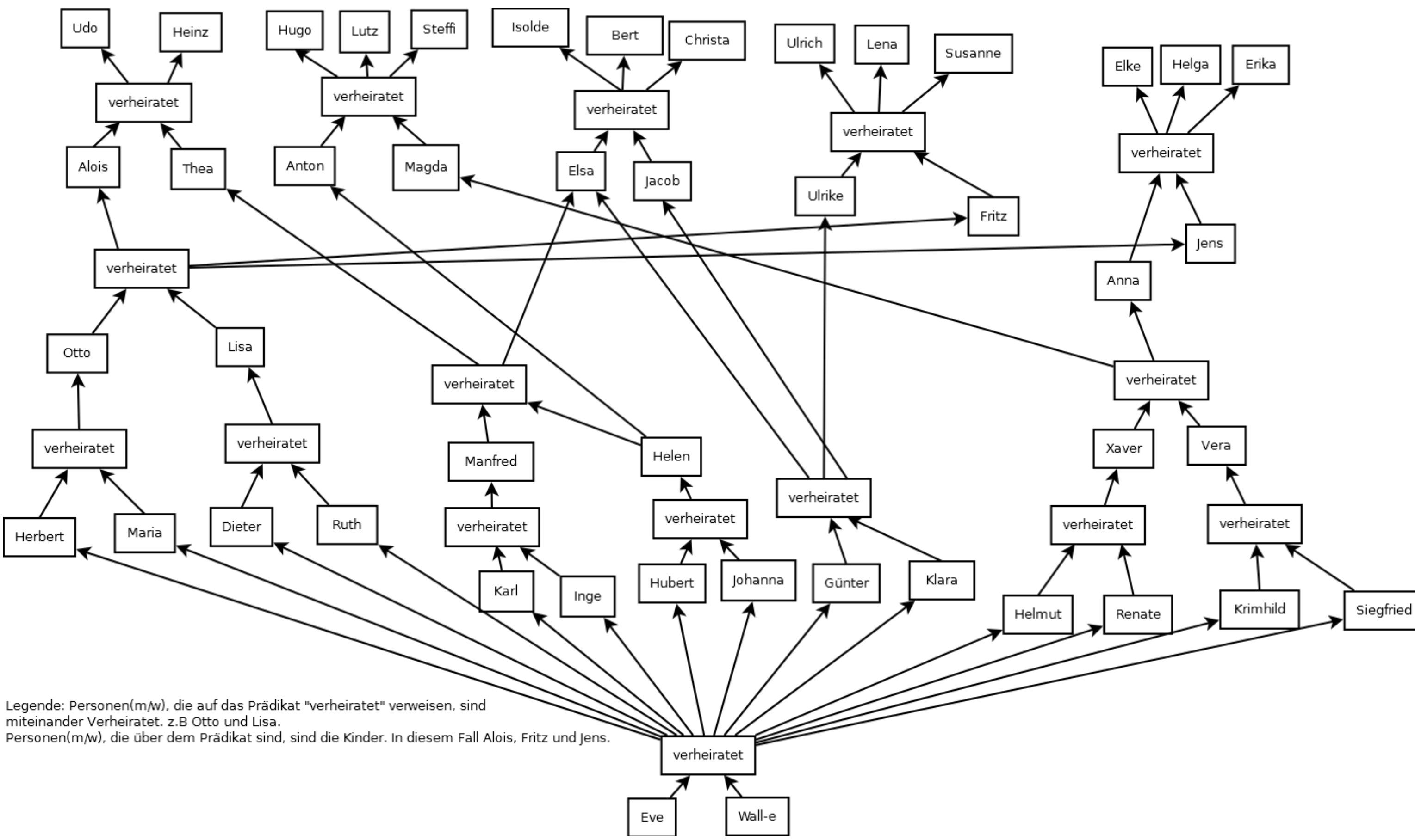
Team: 11, Mesut Koc und Anton Kirakozov

Aufgabenaufteilung: Alle Aufgaben gemeinsam bearbeitet *(Aufgabe_3.pl und Dokumentation)*

Quellenangaben: http://de.wikipedia.org/wiki/Unifikation_(Logik)
Bearbeitungszeitraum: Alle Aufgaben wurden gemeinsam bearbeitet *(siehe gemeinsame Bearbeitungszeit)*
Gemeinsame Bearbeitungszeit: Vom 10.05.15 bis 26.05.15 ca. 18h

Aktueller Stand: Teil 1 und Teil 2 der Aufgabe sind fertig implementiert, zudem sind alle Prädikate funktionsfähig. Die Dokumentation wurde vollständig überarbeitet.

Konzepte unserer fertigen Aufgaben:
Aufgabe 3 Teil 1



Legende: Personen(m/w), die auf das Prädikat "verheiratet" verweisen, sind miteinander Verheiratet. z.B Otto und Lisa.
Personen(m/w), die über dem Prädikat sind, sind die Kinder. In diesem Fall Alois, Fritz und Jens.

3.1.1) In dieser Aufgabe sollen wir eigentlich nur zwei Prädikate definieren *(explizit: **vorfahre**(..) & **nachkomme**(..))*, die jeweils alle Vorfahren und Nachkommen sberechnen. Die Prädikate hierbei sind in einer Relation zu der Familien-Datenbank „familie.pl“.

3.1.2) Die Aufgabe 3.1.2 erfordert das Prädikat **geschwister**(..), dass alle Geschwister von einer bestimmten Person ausgibt. Geschwister können Personen sein, die beispielsweise gleiche Eltern haben.

3.1.3) Die Aufgabe 3.1.3 unterscheidet sich im Wesentlichen nur ganz wenig von der vorherigen Aufgabe. Hierbei sind die Prädikate **bruder**(..) und **schwester**(..) erfordert, die unter der Verwendung der vorherigen Implementation der Funktion geschwister(..) funktionieren sollen.

3.1.4) In dieser Aufgabe und bei dem Prädikat **eheleute**(..), zeigen wir die symmetrische Relation zwischen zwei Eheleuten, das heißt im Grunde genommen, dass wenn *Albert* und *Albertina* Eheleute sind, dann sind ebenfalls *Albertina* und *Albert* Eheleute.

3.1.5) Unter der Verwendung von Verwandtschaftsstufen Oma und Opa, die wir davor definieren, können wir die weiteren Vorfahren von bestimmten Personen feststellen. Anhand dieser Aufgabe kann man deutlich erkennen, dass es eine transitive Relation zwischen Personen vorliegt.

3.1.6) Durch diese Teilaufgabe stellen wir die Konsistenz oder die Inkonsistenz, der uns zur Verfügung gestellten Datenbank *(familie.pl)*. Dabei betrachten wir vier unterschiedliche Fälle; Es darf keine Person mit mehreren Geschlechtern geben. Des Weiteren betrachten wir die Eltern, wobei ein Elternteil aus einer männlichen und weiblichen Person gebildet wird. Außerdem selektieren wir raus, ob es jemanden ohne Eltern gibt. Zusätzlich betrachten wir den Fall, ob alle gespeicherten, verheirateten Personen nach der Bedingung erst männlich und dann weiblich in der richtigen Reihenfolge sind.

Aufgabe 3 Teil 2

Einleitung: In dieser Teilaufgabe ist es erforderlich eine Unifikation durchzuführen und den allgemeinsten Unifikator zu ermitteln. Wir bearbeiten die Aufgaben, in dem wir den Unifikationsalgorithmus anwenden und vergleichen Diese mit der Unifikation, die uns Prolog anbietet. Bei einer Unifikation werden Terme wie z.B. X und a genommen und es wird bestimmt, wie die beiden instanziiert werden müssen, sodass beide den anderen Term darstellen. In unserem Fall wäre bei der Unifikation, dass das X auf das a zeigt und umgekehrt. Durch diese Zuweisung, welche auch Substitution genannt wird erhalten wir genau was wir möchten, also X = a. Wir erkennen, dass zwei Terme unifizieren, wenn es eine Wertzuweisung für Variablen gibt *(Substitution)*. Dabei suchen wir das allgemeinste Unifikat *(MGU)*, da wir keine möglichen Lösungen ausschließen wollen. Anhand unseres Beispiels möchten wir die Schreibweise einer schriftlichen Unifikation festhalten.

(X) (a) → {a/X}

In weiteren Aufgaben werden wir explizit auf die Unifikationen eingehen ohne diese dabei genau zu erläutern.

3.2.1) Der kleinste gemeinsame **M**ost **G**eneral **U**nificator in diesem Fall ist die leere Liste. Wir haben keine Variablen und auch keine Argumente *(atomare Ausdrücke)*, die unifiziert werden können. Somit wird die leere Liste durch eine leere Liste ersetzt.

3.2.2) Bei den beiden Ausdrücken *[X,Y]* und *[c|[[a,b]]]*, kann man erfolgreich eine Unifikation durchführen. Unsere Variablen X und Y werden durch die Argumente der zweiten Liste ersetzt. Somit folgt daraus dass wir zwei MGU's haben, die sich wie folgt definieren; *X* wird durch das *c* ersetzt, da es das erste Element aus der zweiten Liste ist. Der Rest der zweiten Liste also *[a,b]* wird an das *Y* gebunden und *Y* dadurch ersetzt.

3.2.3) Bei unseren nächsten Ausdrücken müssen wir auf einen wesentlichen Faktor achten und zwar ob die Prädikatsfunktionen gleich sind. In diesem Fall trifft es zu und auch hier kann man eine erfolgreiche Unifikation durchführen und den MGU ermitteln. Ebenfalls kommt bei den ermittelten Zeichen eine Variable vor und die Variable X kommt im Term nicht vor, deswegen ist die Unifiktion hier geltend. Da wir in beiden Funktionen jeweils ein Term(t) und eine Variable haben, ersetzen wir einmal das *X* und das *Y*. Das *X* wird mit *r(a,b)* ersetzt und das *Y* durch das *a*.

3.2.4) Betrachten wir nun die beiden Ausdrücke f(X,Y) und f(Y,f(X)), so könnte man laut dem Unifikationsalgorithmus X durch Y ersetzen. Prolog bestätigt, dass die Variablen mit allem unifizieren, was nur denkbar ist. Man kann sagen, dass die beiden Variablen *X* und *Y* auf dasselbe zeigen, möglicherweise auf ein noch nicht existierendes Objekt. Somit würde sich folgender **MGU** ergeben; *X* wird durch *Y* ersetzt und *Y* wird dann durch *f(Y)* ersetzt.

3.2.5) Bei dieser Teilaufgabe, kann <u>keine</u> erfolgreiche Unifikation stattfinden, da die Prädikatsfunktionen sich unterscheiden. *Gegenteil zu 3.2.3.*

3.2.6) In diesem Fall kann eine Unifikation stattfinden. Der gesamte Ausdruck **[1,2|E] - E und [X,Y,F|G] - [a,b,c]** erscheint auf den Ersten Blick etwas verwirrend. Nach längerem durchspielen des Algorithmus kann man erkennen, dass die Variable E und die Liste [a,b,c] als erstes unifiziert werden, wegen dem „-'' Operator. Dieses Minuszeichen wird von Prolog als ein Arithmetischer Ausdruck gewertet und somit gelten auch die Regeln wie Punkt- vor Strichrechnung oder Assoziativität. Wenn der Operator – vor dem Ausdruck steht ist es ein Präfixoperator. Andere Operatoren wie z.B. die Negation oder die Konjunktion also das logische UND haben eine geringere Priorität und werden somit nach anderen Operationen ausgeführt. Gehen wir nun zurück zu unserem Ausdruck.

E wird mit der Liste [a,b,c] ersetzt. Danach können weitere Unifikationen durchgeführt werden, da wir ja keine Operatoren mehr haben, die die Reihenfolge festlegen. Jetzt können wir die beiden übergebliebenen Listen betrachten [1,2IE] und [X,Y,FIG]. Man ersetzt nun das X mit der 1 und das Y mit der 2. Das F wird durch das erste Element von E ersetzt, da sich in der ersten Liste keine weiteren Elemente im Kopf mehr befinden. Zur Erinnerung; E = [a,b,c]. Da aber das F sich im Kopf der zweiten Liste befindet wird auch nur das erste Element aus [a,b,c] gebunden, also das a. Den Rest der E-liste bildet also die Liste [b,c], welche für das G in der zweiten Liste ersetzt wird. Um diese Aussage nochmal etwas deutlicher zu machen möchten wir das nochmal anhand eines erweiterten Beispiels darstellen.

Man betrachte nur die die „Teiliste" [FIG] und [a,b,c] **=>** F = a **=>** [G] und [b,c] **=>** G = [b,c]. Hätten wir nun aber [F,HIG] und [a,b,c] so kommt eine andere Unifikation zustande. F = a => [HIG] und [b,c] . Da das H den Kopf der ersten Liste bildet kriegt es aus der Liste mit [b,c] wieder nur das erste Element also das b, somit folgt H = b. Es bleibt wieder nur [G] und [c], also G= [c]. Die **MGU's** in diesem Fall: E = [a,b,c], X= 1, Y=2, F= a, G= [b,c].

3.2.2) Im letzten Teil der Teilaufgabe 2 entwickeln wir unser eigenes Unifikationsprädikat, welches zwei verschiedene Terme miteinander unifiziert. Dabei verwenden wir keine Hilfsmittel, die uns Prolog anbietet. Unser Prädikat soll eine Variable und ein Argument unifizieren. Anhand eines Beispiels kann man darstellen, wie das Prädikat arbeiten soll:

(X,Y,Z) und (a,b,c) => X=a, Y=b, Z=c.

Um das zu realisieren verwenden wir einen Unifikationsalgorithmus, der in der Vorlesung präsentiert wurde.