Die meisten Programmiersprachen (auch die objekt-orientierten) sind im Wesentlichen *prozedural*. Der Programmierer muss bei diesen Sprachen alle Schritte, die der Computer machen soll, selbst planen und codieren.

Prinzipiell anders ist es bei PROLOG, einer **deklarativen** Sprache. Hierbei muss man dem Computer mit Hilfe von Fakten und Regeln logische Beziehungen definieren, deklarieren oder beschreiben (siehe Beispiele).

Die Sprache PROLOG wurde zu Beginn der 70er Jahre von einer Forschungsgruppe für *Künstliche Intelligenz* in Marseille entwickelt, kann aber auch für einfachere Anwendungen, sogar von Programmier-Anfängern verwendet werden.

# Beispiel 1: Stammbaum einer Familie

In einer Familie bestehen folgende verwandtschaftliche Beziehungen:

August Maria Michael Verena

Josef Paula Burga Andreas Hans Ida Leo

Karl Gerda Anne Walter Denise

Stefanie Christian Corinna Wolfgang Ute

Luca Luis Martin Claudia

Oben stehen die Vorfahren, darunter die Nachkommen.

Offensichtlich gelten hier z. B. folgende Beziehungen:

August ist Vater von Josef. Verena ist Mutter von Ida. Andreas ist Sohn von Michael.

Maria ist Karls Oma. Ida ist Gerdas Tante. Leo ist Annemaries Onkel.

Denise ist Gerdas Cousine und Verenas Enkelin. Maria ist Christians Uroma.

In der Sprache PROLOG ist es relativ leicht, nur notwendigen Fakten und die Regeln einzugeben, wie Verwandtschaftsbeziehungen definiert sind, so dass das Programm dann konkret ausgibt, wer hier wessen Bruder, Onkel, Opa, Cousine, Urgroßenkelin usw. ist:

Öffne das Programm doel\_01\_familie.pl in swish (<https://swish.swi-prolog.org>).

Hinweis: Du kannst im Suchfeld ***user:"doel"*** eingeben. So findest du alle Programme für unsere Übungen.

% FAMILIE.PL Verwandschaftsverhältnisse

% steht am Anfang einzeiliger Kommentare

weiblich(verena).

weiblich(burga).

Fakten und Prädikate enden stets mit einem Punkt.

weiblich(ida).

weiblich(gerda).

weiblich(anne).

weiblich(denise).

männlich(michael).

männlich(andreas).

männlich(leo).

% istElternteilvon(E,X) E ist Elternteil von X

istElternteilvon(verena,andreas).

istElternteilvon(verena,ida).

istElternteilvon(michael,andreas).

istElternteilvon(michael,ida).

istElternteilvon(burga,gerda).

istElternteilvon(andreas,gerda).

istElternteilvon(burga,annemarie).

istElternteilvon(andreas,annemarie).

istElternteilvon(ida,denise).

istElternteilvon(leo,denise).

:- kann als „wenn“ gelesen werden.

Die Kommas stehen für „UND“

% istBrudervon(B,X): B ist Bruder von X

istBrudervon(B,X) :-

männlich(B),

\ = = heißt

„ungleich“

istElternteilvon(E,B),

istElternteilvon(E,X),

X\==B.

% istSchwestervon(S,X): S ist Schwester von X

istSchwestervon(S,X) :-

weiblich(S),

istMuttervon(E,S),

istMuttervon(E,X),

X\==S.

% istMuttervon(M,K) M ist Mutter von Kind K

istMuttervon(M,K) :-

weiblich(M),

istElternteilvon(M,K).

% istOmavon(O,E) O ist Oma von Enkel(-in) E

istOmavon(O,E) :-

istMuttervon(O,X),

istElternteilvon(X,E).

**Aufgabe 1:**

Mit diesen Informationen kann das Programm schon einiges anfangen. Man kann z. B. fragen, ob Andreas Elternteil von Denise ist.

Gib dafür rechts unten hinter ?- ein: istElternteilvon(andreas,denise). (Achte auf die ***klein***geschriebenen Namen!)

Das Programm antwortet lapidar mit No. Andreas ist also *kein* Elternteil von Denise.

**Aufgabe 2**: Teste, ob bestimmte Personen auf eine bestimmte Art miteinander verwandt sind oder nicht. Verwende dazu die bereits fertigen Regel. Provoziere dabei dreimal No und dreimal Yes.

No liefert: a) istElternteilvon(gerda,denise)

b) istElternteilvon(andreas,leo) c) istElternteilvon(ida,michael)

Yes liefert: a) istElternteilvon(leo,denise) b) istElternteilvon(burga,gerda) c) istElternteilvon(verena,ida)

**Aufgabe 3**: Gib als Abfrage männlich(Verena) ein. (Verena groß geschrieben!)

Wie ist diese Reaktion zu erklären? Lass dir alle Antworten anzeigen.

Merke: Groß geschriebene Wörter interpretiert PROLOG alsVariable\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

**Aufgabe 4**: Vergleiche die Antworten auf istBrudervon(B,X) mit den Antworten auf

istSchwestervon(S,X). Lass dir wieder alle Antworten anzeigen.

Was fällt auf?

Bei istBrudervon kommt bei B nur andreas und bei X nur ida, bei istSchwestervon kommt zuerst S = ida X = andreas und dann S = gerda X = annemarie

**Aufgabe 5**: Erweitere FAMILIE.PL um die Regeln für weitere Verwandtschaftsbeziehungen: istVatervon(V,K), istOpavon(O,E), ist Sohnvon(S,E), istTochtervon(T,E), istTantevon(T,N), istOnkelvon(O,N), istUrenkelvon(U,V) und bei gutem Fortschritt noch mehr Regeln (siehe Beziehungen unter Stammbaum, S. 1)

Sonderzeichen „\_“

Für ganz Uninteressantes gibt es das Sonderzeichen „\_“ (Unterstrich).

Man verwendet es z. B. bei der Frage nach allen Vätern, wenn man gar nicht wissen will, wie die zugehörigen Kinder heißen.

**Aufgabe 6**: Frage nach istVatervon(X,Y) und danach istSohnvon(X, \_ ).

Verknüpfungen in Anfragen

Nicht nur in der Wissensbasis können Aussagen mit and (dafür schreiben wir ein Komma) und or (;) verknüpft werden (beachte die höhere Priorität von and als von or), sondern auch erst in Abfragen:

Frage z. B. nach allen Müttern, die Töchter haben:

istMuttervon(X,Y), weiblich(Y).

**Aufgabe 7**: Frage nach allen Cousinen von Frauen:

Eingabe:diese Übung hab ich nicht mehr geschafft

Ausgabe:

Finde eigene Beispiele für verknüpfte Anfragen: