IV. Logische Programmierung

- **1. Grundkonzepte der logischen Programmierung**
- **2. Syntax von Prolog**
- **3. Rechnen in Prolog**

Übersicht

Imperative Sprachen

- Folge von nacheinander ausgeführten Anweisungen

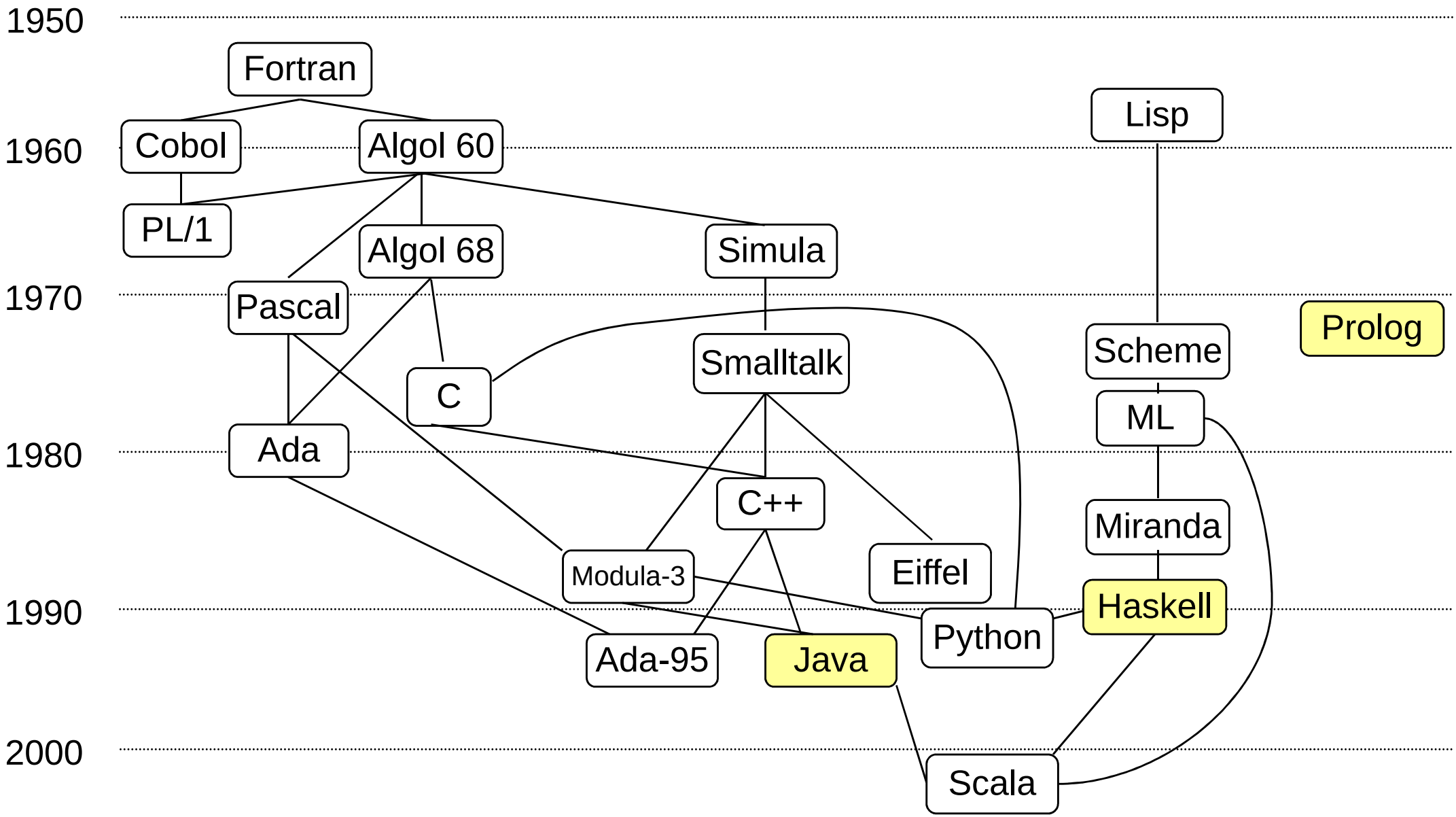
Deklarative Sprachen

- Spezifikation dessen, *was* berechnet werden soll
- Compiler legt fest, wie Berechnung verläuft

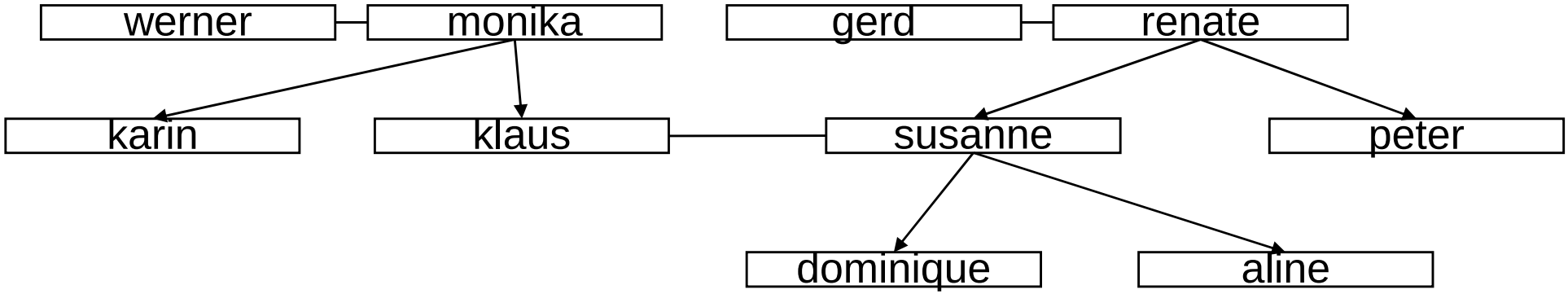
- **Prozedurale Sprachen**
  - Variablen, Zuweisungen, Kontrollstrukturen
- **Objektorientierte Sprachen**
  - Objekte und Klassen
  - ADT und Vererbung

- **Funktionale Sprachen**
  - keine Seiteneffekte
  - Rekursion
- **Logische Sprachen**
  - Regeln zur Definition von Relationen

Wichtige Programmiersprachen



Fakten und Anfragen



Programm:

```
weiblich(monika).
weiblich(karin).
weiblich(renate).
weiblich(susanne).
weiblich(aline).

verheiratet(werner, monika).
verheiratet(gerd, renate).
verheiratet(klaus, susanne).

mensch(X).
```

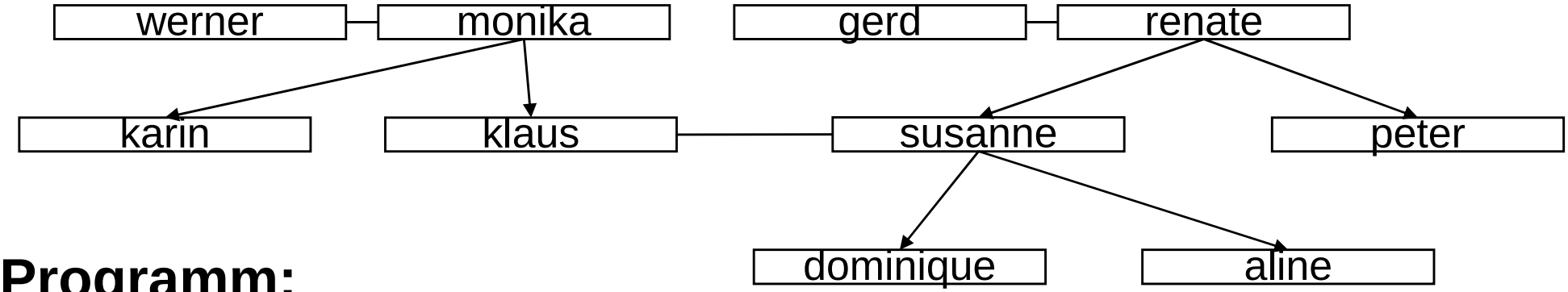
```
maennlich(werner).
maennlich(klaus).
maennlich(gerd).
maennlich(peter).
maennlich(dominique).

mutterVon(monika, karin).
mutterVon(monika, klaus).
mutterVon(renate, susanne).
mutterVon(renate, peter).
mutterVon(susanne, aline).
mutterVon(susanne, dominique).
```

```
?- maennlich(gerd).
true
```

```
?- verheiratet(gerd, monika).
false
```

```
?- mensch(gerd).
true
```

Variablen in Anfragen



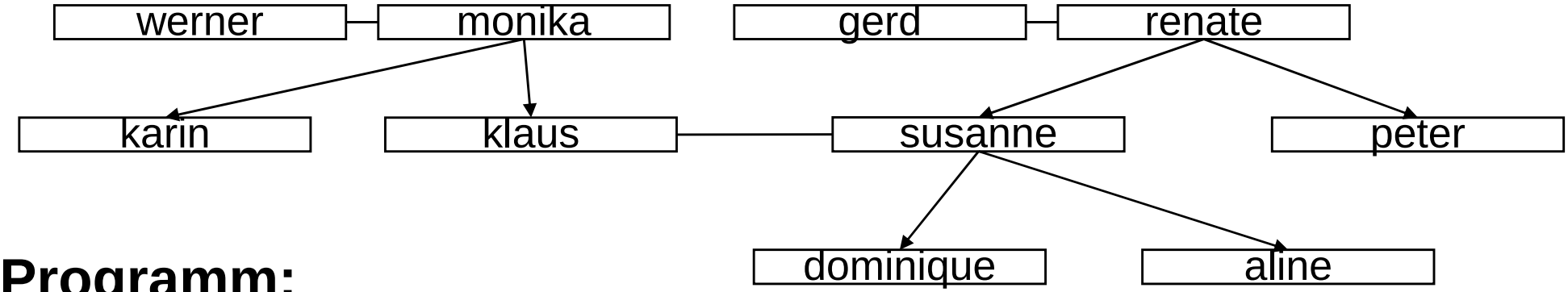
Programm:

```
weiblich(monika).                    maennlich(werner).
...                                  ...
weiblich(aline).                     maennlich(dominique).
verheiratet(werner, monika).         mutterVon(monika, karin).
...                                  ...
verheiratet(klaus, susanne).         mutterVon(susanne, dominique).
```

```
?- mutterVon(X, susanne).
X = renate
```

```
?- mutterVon(renate, Y).
Y = susanne ;
Y = peter
```

Kombination von Fragen

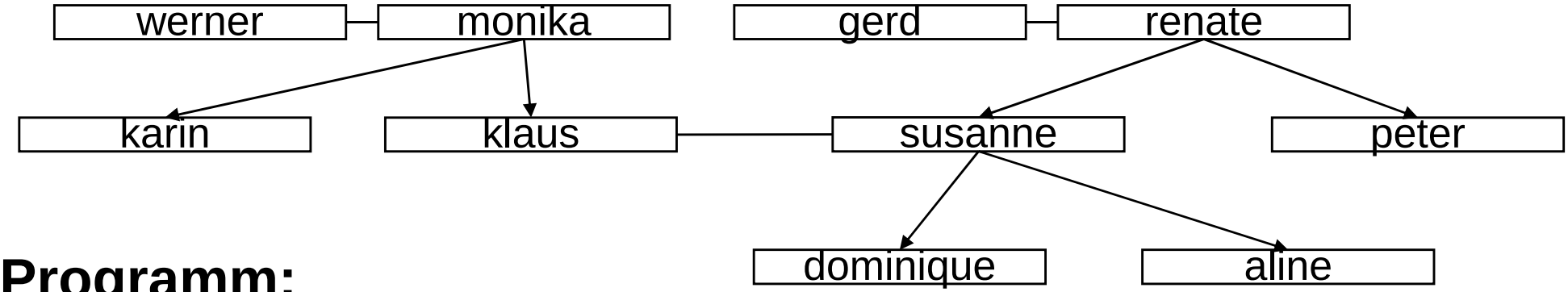


Programm:

```
weiblich(monika).                    maennlich(werner).
...                                  ...
weiblich(aline).                     maennlich(dominique).
verheiratet(werner, monika).         mutterVon(monika, karin).
...                                  ...
verheiratet(klaus, susanne).         mutterVon(susanne, dominique).
```

```
?- verheiratet(gerd,F), mutterVon(F,susanne).
F = renate

?- mutterVon(Oma,Mama), mutterVon(Mama,aline).
Oma = renate
Mama = susanne
```

Regeln



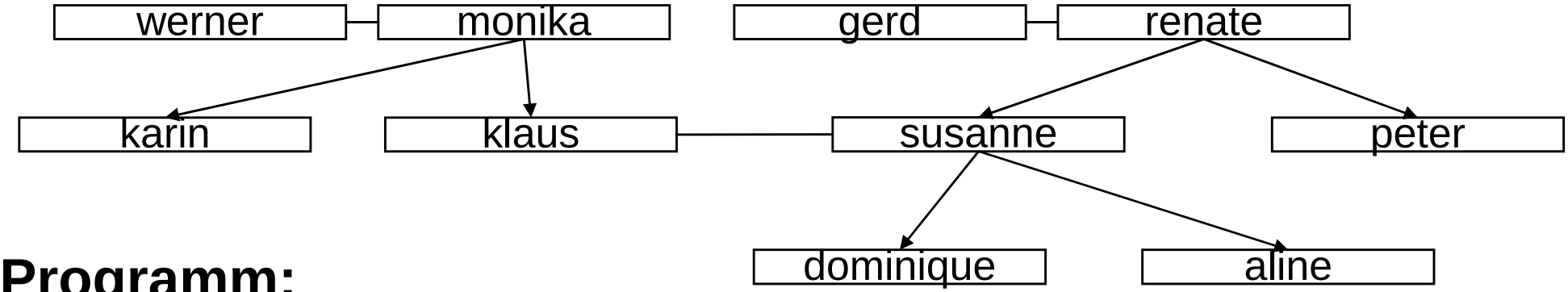
Programm:

```
weiblich(monika).                    maennlich(werner).
...                                  ...
weiblich(aline).                     maennlich(dominique).
verheiratet(werner, monika).         mutterVon(monika, karin).
...                                  ...
verheiratet(klaus, susanne).         mutterVon(susanne, dominique).

vaterVon(V,K)    :- verheiratet(V,F), mutterVon(F,K).
```

```
?- vaterVon(gerd, susanne).
true
```

```
?- vaterVon(gerd, Y).
Y = susanne ;
Y = peter
```

Mehrere Regeln für ein Prädikat

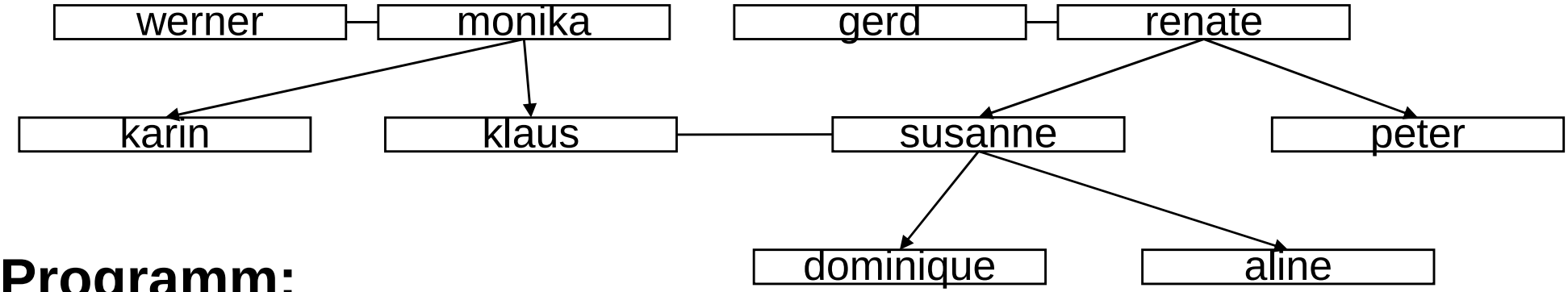


Programm:

```
weiblich(monika).                    maennlich(werner).
...                                  ...
weiblich(aline).                     maennlich(dominique).
verheiratet(werner, monika).         mutterVon(monika, karin).
...                                  ...
verheiratet(klaus, susanne).         mutterVon(susanne, dominique).
vaterVon(V,K)    :- verheiratet(V,F), mutterVon(F,K).

elternteil(X, Y)     :- mutterVon(X,Y).
elternteil(X, Y)     :- vaterVon(X,Y).
```

```
?- elternteil(X, susanne).

X = renate ;

X = gerd
```

Rekursive Regeln



Programm:

```
weiblich(monika).                    maennlich(werner).
...                                  ...
verheiratet(klaus, susanne).         mutterVon(susanne, dominique).
vaterVon(V,K)    :- verheiratet(V,F), mutterVon(F,K).

elternteil(X, Y)    :- mutterVon(X,Y).
elternteil(X, Y)    :- vaterVon(X,Y).

vorfahre(V,X) :- elternteil(V,X).
vorfahre(V,X) :- elternteil(V,Y), vorfahre(Y,X).
```

```
?- vorfahre(X, aline).

X = susanne ;   X = klaus ;
X = monika  ;   X = renate;   X = werner ;   X = gerd
```

Kennzeichen logischer Programme

- **Programme = Fakten und Regeln**
- **Keine Kontrollstrukturen**
- **Ein- und Ausgabevariablen liegen nicht fest**
- **Besonders gut geeignet für Künstliche Intelligenz (z.B. *Expertensysteme, deduktive Datenbanken*)**