

---

# IV. Logische Programmierung

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

# Übersicht

---

## Imperative Sprachen

- Folge von nacheinander ausgeführten Anweisungen

### ■ Prozedurale Sprachen

- Variablen, Zuweisungen, Kontrollstrukturen

### ■ Objektorientierte Sprachen

- Objekte und Klassen
- ADT und Vererbung

## Deklarative Sprachen

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

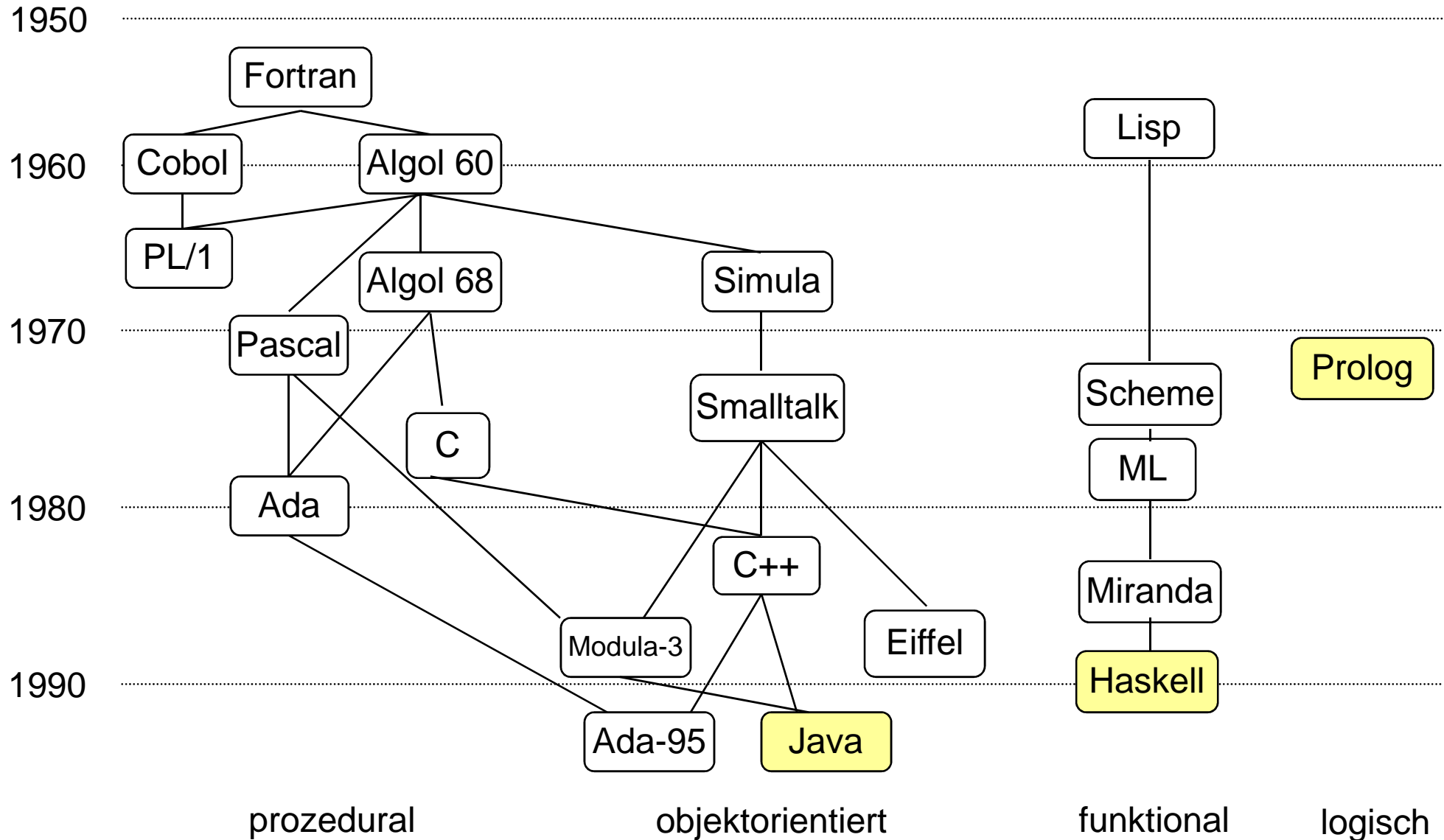
### ■ Funktionale Sprachen

- keine Seiteneffekte
- Rekursion

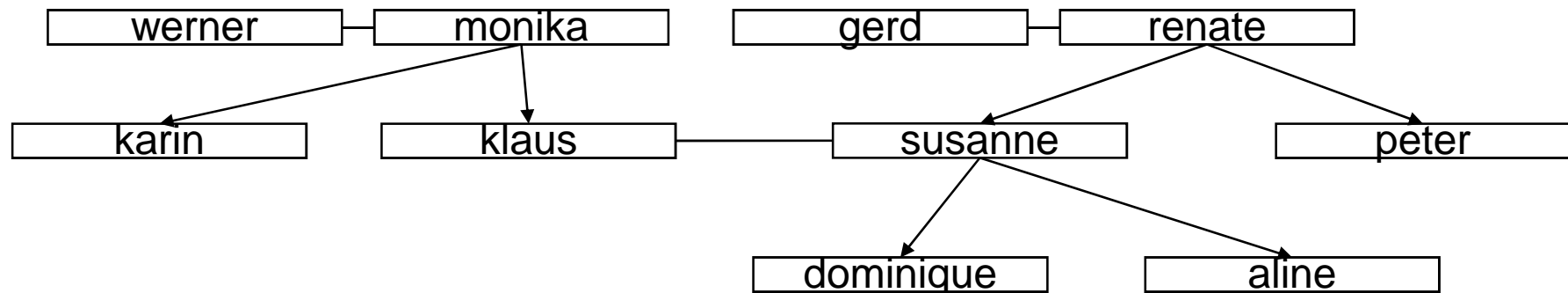
### ■ Logische Sprachen

- 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, reate).  
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).
```

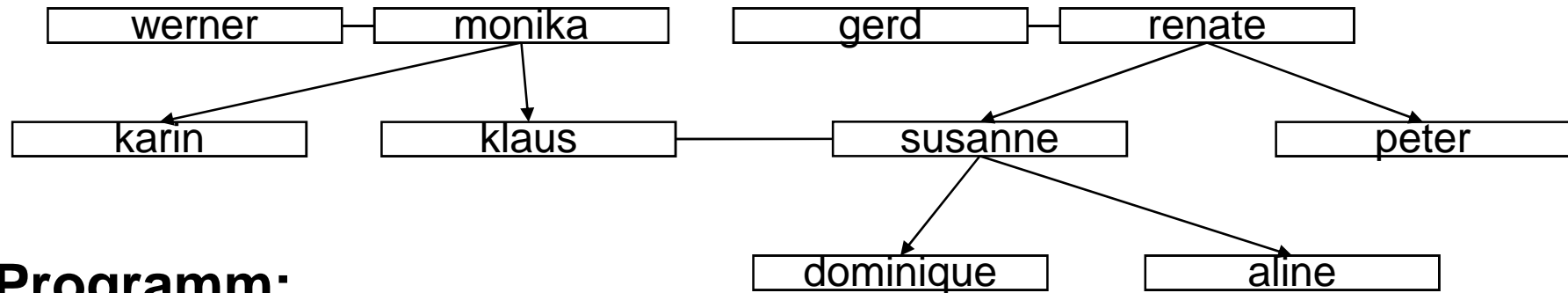
?- maennlich(gerd).    ?- verheiratet(gerd, monika).    ?- mensch(gerd).

**true**

**false**

**true**

# Variablen in Anfragen



## Programm:

```
weiblich(monika).
```

```
weiblich(aline).
```

```
verheiratet(werner, monika).
```

```
verheiratet(klaus, susanne).
```

```
maennlich(werner).
```

```
maennlich(dominique).
```

```
mutterVon(monika, karin).
```

```
mutterVon(susanne, dominique).
```

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

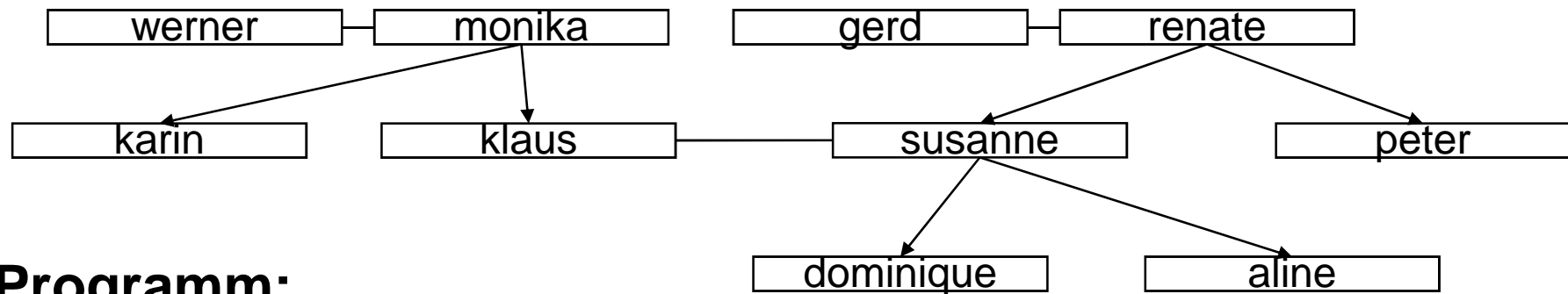
```
X = reate
```

```
?- mutterVon(renate, Y).
```

```
Y = susanne ;
```

```
Y = peter
```

# Kombination von Fragen



## Programm:

```
weiblich(monika).
```

```
weiblich(aline).
```

```
verheiratet(werner, monika).
```

```
verheiratet(klaus, susanne).
```

```
maennlich(werner).
```

```
maennlich(dominique).
```

```
mutterVon(monika, karin).
```

```
mutterVon(susanne, dominique).
```

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

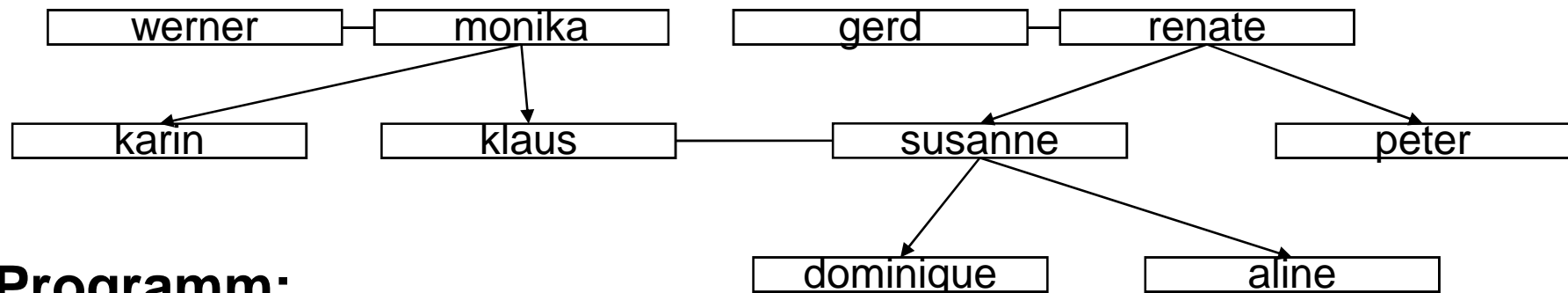
**F = reate**

```
?- mutterVon(Oma,Mama), mutterVon(Mama,aline).
```

**Oma = reate**

**Mama = susanne**

# Regeln



## Programm:

```
weiblich(monika).
```

```
weiblich(aline).
```

```
verheiratet(werner, monika).
```

```
verheiratet(klaus, susanne).
```

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

```
maennlich(werner).
```

```
maennlich(dominique).
```

```
mutterVon(monika, karin).
```

```
mutterVon(susanne, dominique).
```

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

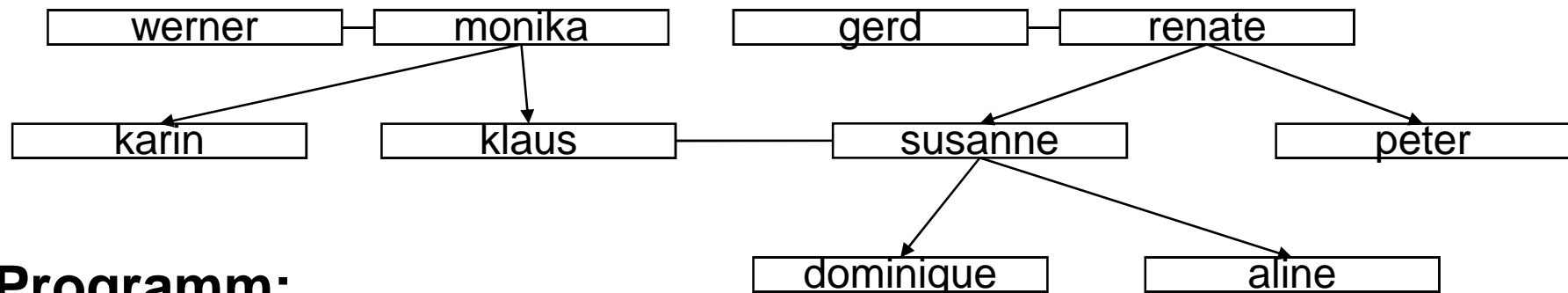
```
true
```

```
?- vaterVon(gerd, Y).
```

```
Y = susanne ;
```

```
Y = peter
```

# Mehrere Regeln für ein Prädikat



## Programm:

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

maennlich(werner).
maennlich(dominique).
mutterVon(monika, karin).
mutterVon(susanne, dominique).

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

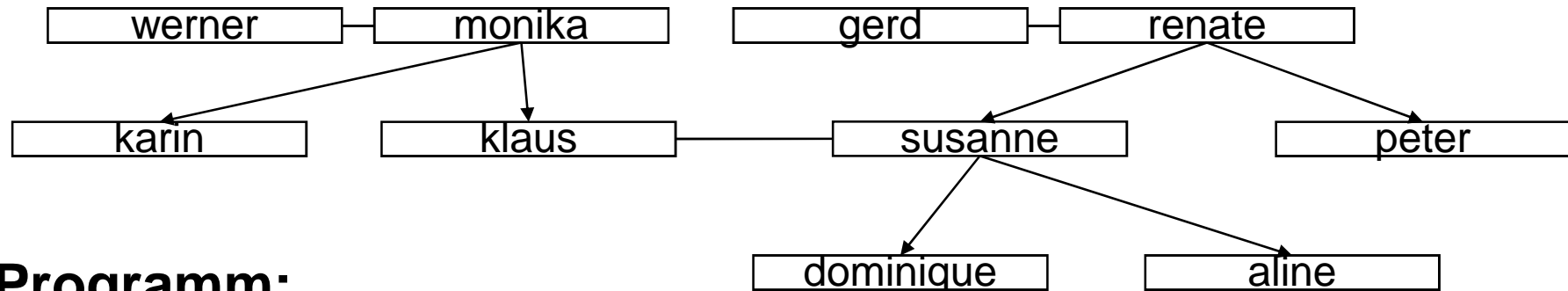
```
?- elternteil(X, susanne).
```

```
X = renete ;
```

```
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 = reate ; 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*)