

Mathematik 1 für Informatiker



Grundbegriffe

Einführung in die
Logik u. Algebra

27.10.2006

M. B. Wischnewsky

Produkt von Mengen

Definition Sei $(A_i, i \in I)$ eine Familie von Mengen.

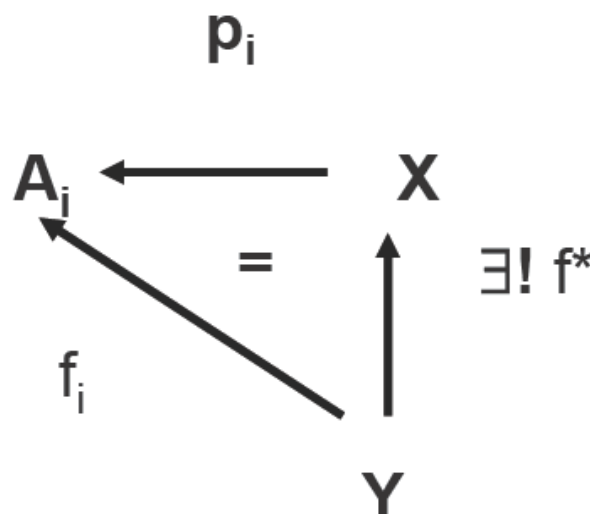
Eine Menge X zusammen mit einer Familie von Abbildungen

$p_i: X \rightarrow A_i$ heißt **Produkt** der $(A_i, i \in I)$, wenn folgendes gilt:

Zu jeder Menge Y und jeder Familie von Abbildungen

$f_i: Y \rightarrow A_i$ existiert genau eine Abbildung $f^*: Y \rightarrow X$ mit

$$\forall i \in I \quad p_i \circ f^* = f_i$$



Produkt von Mengen

Definition Sei $(A_i, i \in I)$ eine Familie von Mengen.

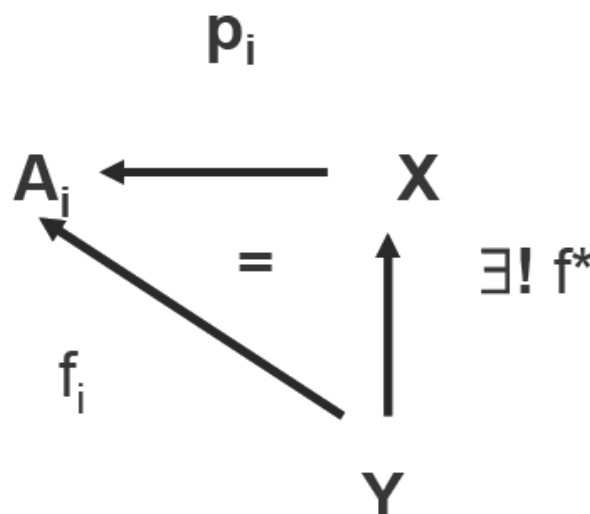
Eine Menge X zusammen mit einer Familie von Abbildungen

$p_i: X \rightarrow A_i$ heißt **Produkt** der $(A_i, i \in I)$, wenn folgendes gilt:

Zu jeder Menge Y und jeder Familie von Abbildungen

$f_i: Y \rightarrow A_i$ existiert genau eine Abbildung $f^*: Y \rightarrow X$ mit

$$\forall i \in I \quad p_i \circ f^* = f_i$$



Relationen

Definition

Eine **n-stellige Relation R** ist eine Teilmenge des (kartesischen) Produktes von n Mengen A_1, A_2, \dots, A_n :

$$R \subseteq A_1 \times A_2 \times \dots \times A_n$$

Schreibweise:

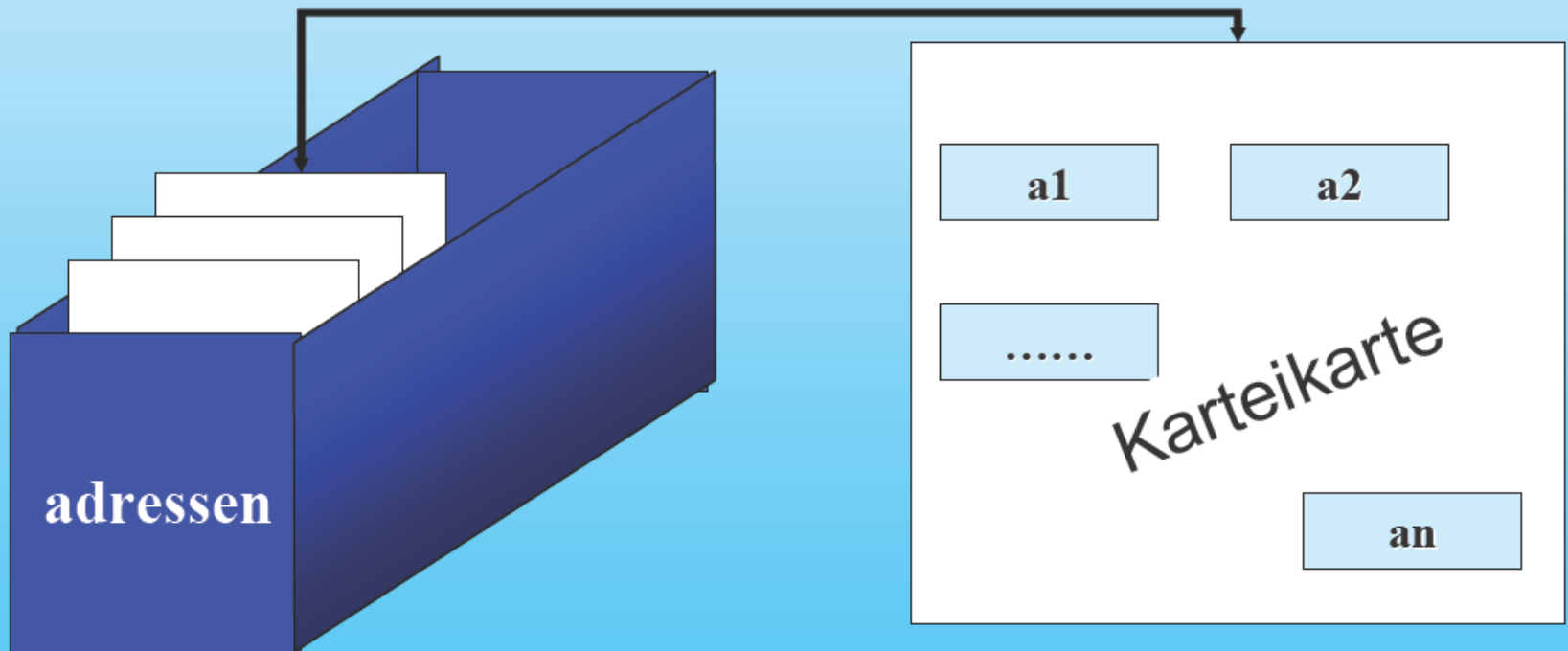
$$(a_1, a_2, \dots, a_n) \in R \leftrightarrow R(a_1, a_2, \dots, a_n)$$

Relationen

Beispiele

1. Datenbanken: Tabelle

Beispiel: $\text{adressen}(a_1, a_2, \dots, a_n)$



4 Programmierparadigmen

- Imperatives Programmieren
- Funktionales Programmieren
- Deklaratives Programmieren
- Objektorientiertes Programmieren

Programmlängen-Vergleich

<u>Programmiersprache</u>	<u>Programmgröße in Quellseiten</u>
Fortran	36
Cobol	25
Ada	24
PL/I	22
C	22
Pascal	20
Basic	19
MProlog	9

1. Imperatives Programmieren

Es wird beschrieben, **WIE** ein bestimmtes Problem gelöst werden soll.

- Assembler
- ADA
- BASIC
- C / C++
- COBOL
- FORTRAN
- **Java**
- Modula
- PASCAL
- Perl
- PL/1
- Simula
- Smalltalk
- und viele mehr...

2. Funktionales Programmieren

Das Programm ist eine Folge von Funktionen.

- **Lisp**
- Logo
- **Haskell**
- ML
- Hope
- Scheme
- Concurrent
Clean
- Erlang
- NESL
- Sisal
- **Miranda**

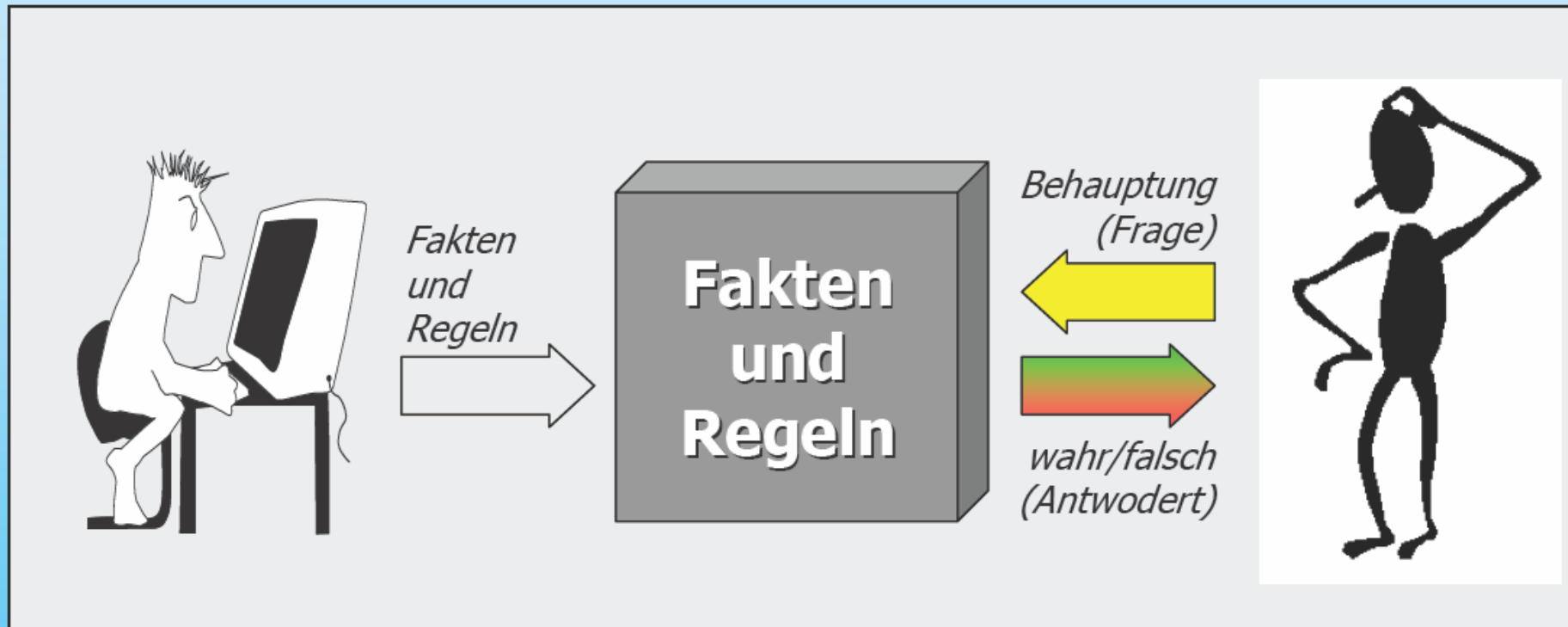
3. Deklaratives Programmieren

Es wird beschrieben, **WAS** das Problem ist, nicht jedoch wie dieses zu lösen ist. Die Lösung muß der Computer finden.

- **Prolog**
- Goedel
- Escher
- Elf
- Mercury

Für was steht PROLOG?





**Prolog-
Programmierer**

**Prolog-
Programiersystem**

Benutzer

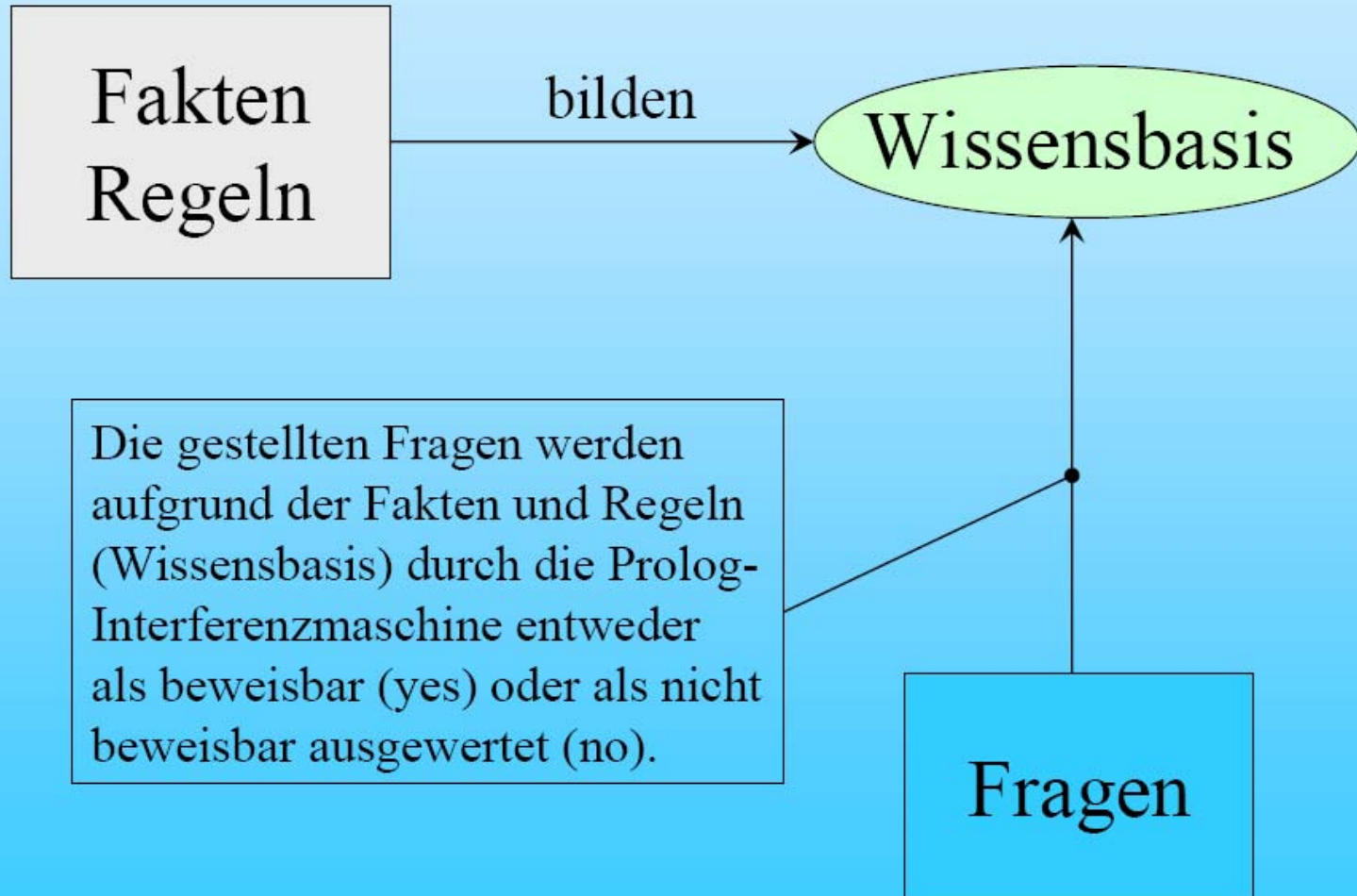
Anwendungsgebiete

- **Expertensystem (Diagnosesystem)**
- **Relationale Datenbanken**
- **mathematische Logik**
- **abstrakte Problemlösungen**
- **Simulieren des menschlichen Sprachverstehens**
- **automatischer Entwurf**
- **Lösung von symbolischen Gleichungen**
- **Analyse biochemischer Strukturen**
- **zahlreiche Gebiete der KI**

Geschichte der Logischen Programmierung

Die Geschichte der Logischen Programmierung ist nicht sehr lang. Die theoretischen Grundlagen wurden in den 70er Jahren erarbeitet. Der erste Prolog-Interpreter wurde 1972 von Alain Colmerauer in Marseilles geschrieben. Erst Anfang der 80er Jahre kamen die ersten kommerziellen Prolog-Interpreter auf den Markt. Durch die Wahl von PROLOG als Sprache der Rechner der 5. Generation bei einem jap. Forschungsprojekt gelang der weltweite Durchbruch.

Das Prinzip von PROLOG

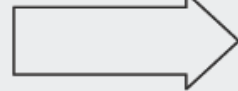


PROLOG

PROgrammieren in LOGig



*Fakten
und
Regeln*



**Fakten
und
Regeln**

*Behauptung
(Frage)*



*wahr/falsch
(Antwort)*



**Prolog-
Programmierer**

**Prolog-
Programiersystem**

Benutzer

Fakten 2

Beispiele:

Schreibweise in Prolog:

- 'die Sonne scheint'.
- es_regnet.
- mensch(sokrates).
- männlich(daniel).
- mag(johann,maria).
- besitzt(johann,gold).
- vater(hans, gabriel).

Natürliche Bedeutung:

- Die Sonne scheint.
- Es regnet.
- Sokrates ist ein Mensch.
- Daniel ist männlich.
- Johann mag Maria.
- Johann besitzt Gold.
- Hans ist der Vater von Gabriel.