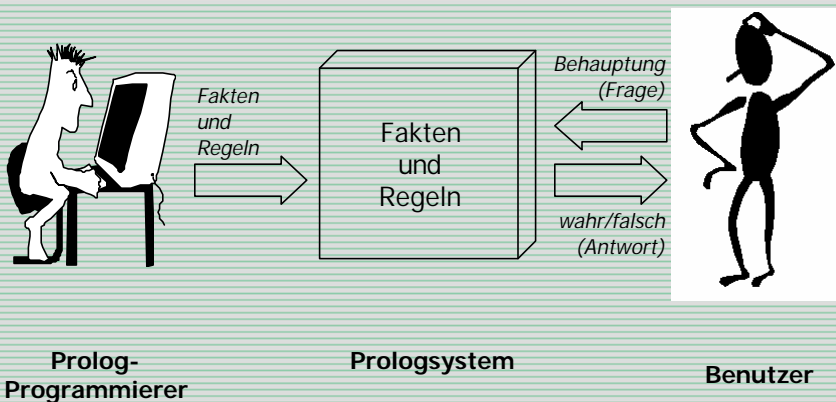


Spezielle Themen der KI

Eine Einführung in Prolog und die logische Programmierung

Marc Erich Latoschik, nach Marla Bachman
AI & VR Lab, Faculty of Technology
University of Bielefeld
marcl@techfak.uni-bielefeld.de

Programmieren in Prolog



Programmieren in Prolog

Programmieren in Prolog bedeutet

- **Tatsachen (Fakten)** über Objekte und deren Beziehungen zu deklarieren *Sokrates ist ein Mensch.*
- **Regeln** über Objekte und deren Beziehungen zu definieren *Alle Menschen sind sterblich.*
- **Fragen** zu den Objekten und Beziehungen zu stellen *Ist Sokrates sterblich?*

3

Prozedurale vs logische (deklarative) Programmierung

Prozedural

Der Benutzer sagt, *wie* das Problem gelöst werden soll.

Beschreibung eines Kreises:
Resultat einer 360 Grad Rotation mit dem Zirkel.

Deklarativ

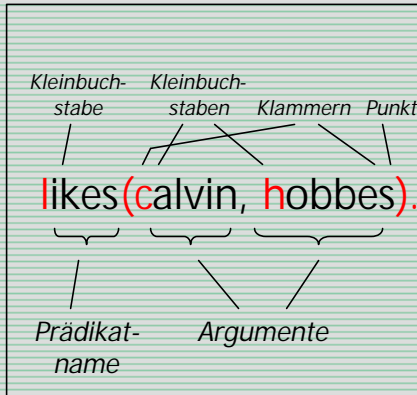
Der Benutzer beschreibt, *was* das Problem ist. *Wie* es gelöst wird, wird vom System kontrolliert.

Beschreibung eines Kreises:
Menge aller Punkte, die von einem vorgegebenen Punkt denselben Abstand hat.

4

Prolog Essentials

Fakten



Weitere Beispiele von Fakten:

`tiger(hobbes).`
`person(calvin).`
`dreams(calvin).`
`lazy(calvin).`
`gives(calvin, ball, hobbes).`
 ...

Prolog Essentials

Fragen

?- likes(calvin, hobbes).

yes

?- tiger(hobbes).

yes

?- tiger(calvin).

no

?- lion(calvin).

no

`tiger(hobbes).`

`person(calvin).`

`person(susie).`

`likes(calvin, hobbes).`

`likes(susie, school).`

`likes(susie, hobbes).`

Prolog Essentials

Variablen

?- likes(calvin, X).

X=hobbes;

No

Grossbuch-
staben

?- likes(**W**ho, hobbes).

Who=calvin;

Who=susie;

no

tiger(hobbes).

person(calvin).

person(susie).

likes(calvin, hobbes).

likes(susie, school).

likes(susie, hobbes).

Prolog Essentials

Konjunktionen

Mögen sich Calvin und Hobbes gegenseitig?

?- likes(calvin, hobbes), likes(hobbes, calvin).

„und“

Gibt es etwas, das sowohl Calvin als auch Susie mögen?

?- likes(susie, X), likes(calvin, X).

tiger(hobbes).

person(calvin).

person(susie).

likes(calvin, hobbes).

likes(susie, school).

likes(susie, hobbes).

Prolog Essentials

Beantwortungsmechanismus (1)

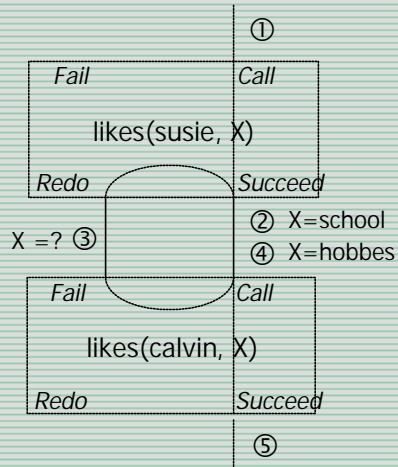
```
tiger(hobbes).
person(calvin).
person(susie).

likes(calvin, hobbes).
likes(susie, school).
likes(susie, hobbes).
```

?- likes(susie, X), likes(calvin, X).

X=hobbes;
no

zu beweisende
Goals



Prolog Essentials

Regeln

„falls“

```
Alle Personen mögen Hobbes.
likes(X, hobbes) :- person(X).
```

Head
Body

„Ein Objekt mag Hobbes, falls dieses Objekt eine Person ist.“

Weitere Beispiele von Regeln:

```
likes(X, hobbes) :- person(X), nice(X).
```

```
likes(X, hobbes) :- tiger(X).
```

```
may_steal(Person, Thing) :- thief(Person), likes(Person, Thing).
```

Prolog Essentials

Beantwortungsmechanismus (2)

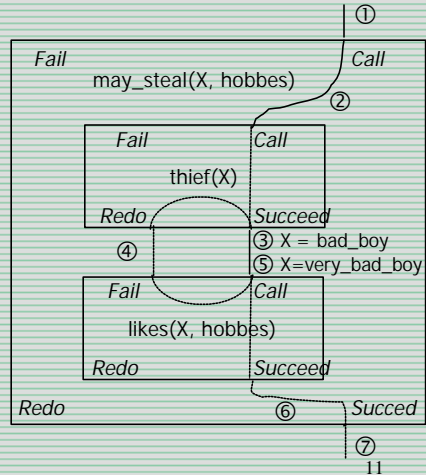
```

thief(bad_boy).
thief(very_bad_boy).

likes(calvin, hobbes).
likes(bad_boy, money).
likes(very_bad_boy, hobbes).

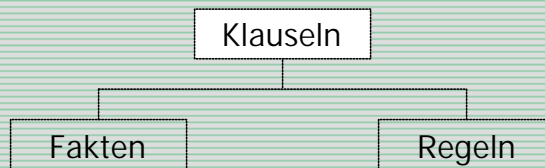
may_steal(Person, Thing) :-
    thief(Person),
    likes(Person, Thing).
    
```

?- may_steal(X, hobbes).
X=very_bad_boy



Prolog Essentials

Klauseln



tiger(hobbes).
likes(calvin, hobbes).
...

likes(X, hobbes) :- person(X).
egoist(X) :- person(X), likes(X, X).
...

Prolog Essentials

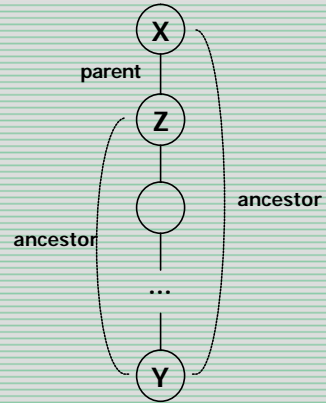
Rekursive Regeln (1)

```

ancestor(X, Y) :-
    parent(X, Y).
ancestor(X, Y) :-
    parent(X, Z),
    ancestor(Z, Y).
    
```

Abbruchbedingung

Rekursion



Eine Prozedur ist *rekursiv*, wenn in einer Regel das Prädikat, durch das die Prozedur definiert ist, wieder aufgerufen wird.

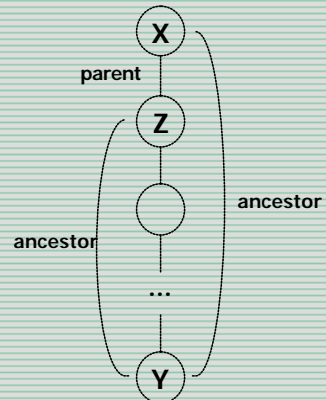
Prolog Essentials

Rekursive Regeln (2)

```

ancestor(X, Y) :-
    ancestor(Z, Y),
    parent(X, Z).
ancestor(X, Y) :-
    parent(X, Y).
    
```

?



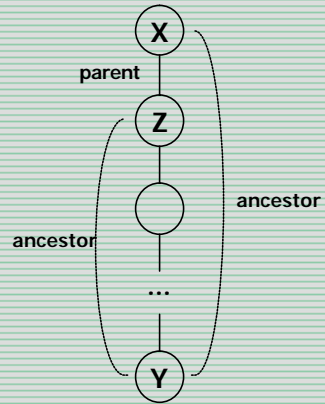
Prolog Essentials

Rekursive Regeln (3)

```

ancestor(X, Y) :-
    ancestor(Z, Y),
    parent(X, Z).
ancestor(X, Y) :-
    parent(X, Y).
    
```

Die Prozedur ist vom deklarativen Standpunkt aus korrekt. Sie führt aber mit der Prolog-Auswertungsstrategie zu einer Endlosschleife



Prozedurale Komponente von Prolog

$H :- B1, B2.$

Deklarative Semantik

H erfüllt, falls sowohl B1 als auch B2 erfüllt werden können

Prozedurale Semantik

H erfüllt, falls zuerst B1 und dann B2 erfüllt werden können

Für die Abarbeitung der Prolog-Programme wird den Programmklauseln eine prozedurale Semantik unterlegt.