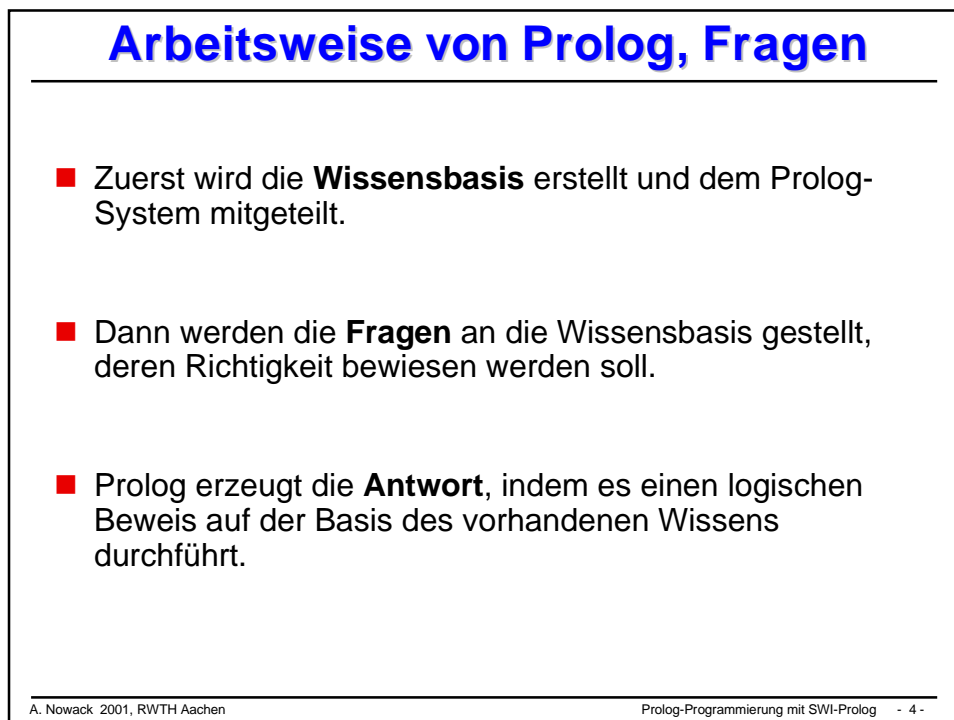
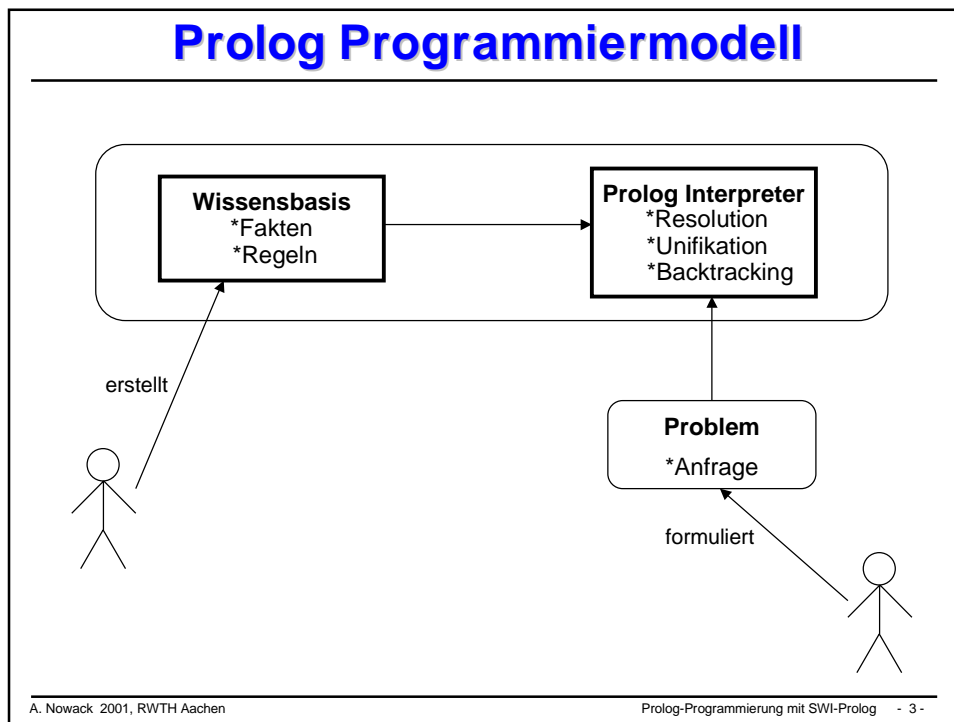


Prolog Programmierung mit SWI Prolog

- SWI Prolog
- Arbeiten mit SWI Prolog
- Laden einer Datei
- Fehler
- Zusammenfassung

SWI Prolog

- Prolog
 - steht für **Programming in Logic**
 - Logik wird als Programmiersprache benutzt, basiert auf der **Prädikatenlogik**
 - ist gut geeignet für Probleme, die Objekte und Relationen zwischen Objekten behandeln
 - Syntax der Sprache ist eine modifizierte **Hornklausel-Notation**
- SWI Prolog
 - ist freie Software und unter GNU-Lizenz frei verteilbar
 - besteht aus einem Interpreter



Fakten, Regeln, Programme (1)

- **Regeln** dienen dazu, aus bekanntem Wissen neues Wissen herzuleiten

- **Regeln** bestehen aus
 - einem Bedingungsteil (rechte Seite, Rumpf)
 - und einem Folgerungsteil (linke Seite, Kopf)

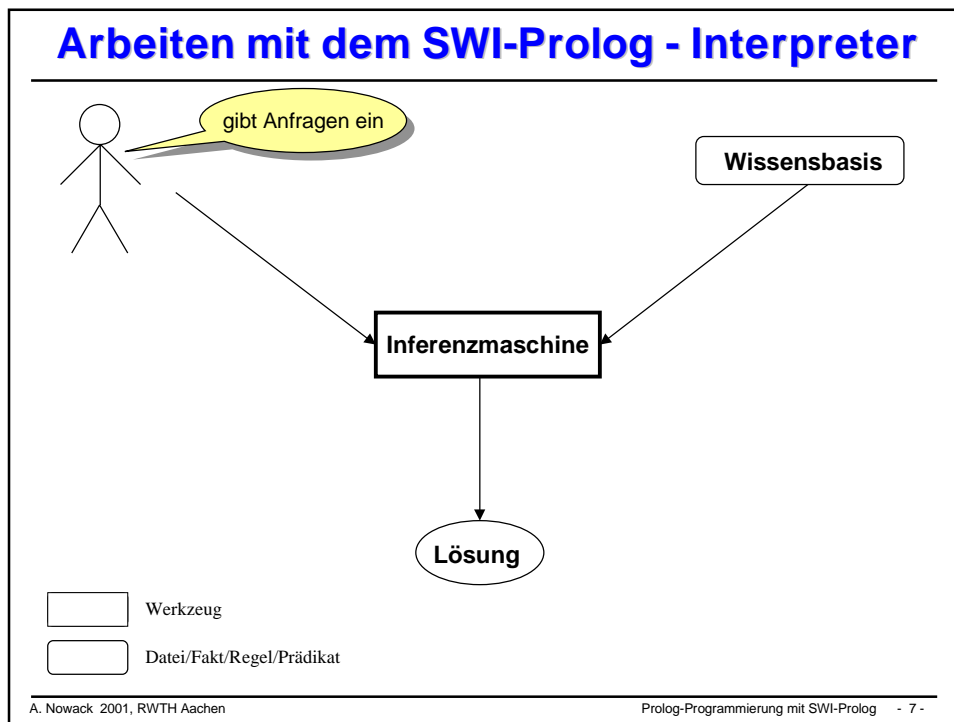
- Eine Regel **p:-q, r** kann gelesen werden
 - Deklarative Bedeutung im Sinne von WAS
 - ◆ IF rechte Seite der Regel THEN linke Seite der Regel
 - Prozedurale Bedeutung im Sinne von WIE
 - ◆ Um ein Ziel p zu lösen, müssen zuerst die Unterziele q und r gelöst werden

Fakten, Regeln, Programme (2)

- Prolog wendet die Regeln rückwärts an
 - d.h. Prolog startet mit der linken Seite
 - um zu zeigen, daß die linke Seite gilt, muß gezeigt werden, daß die rechte Seite gilt

- Prolog-**Programme** bestehen aus Fakten und Regeln

- Treten Variablen in Fragen auf, so antwortet Prolog
 - mit einer Instanziierung der Variablen in der Frage
 - weitere Instanziierungen können mit ; abgerufen werden



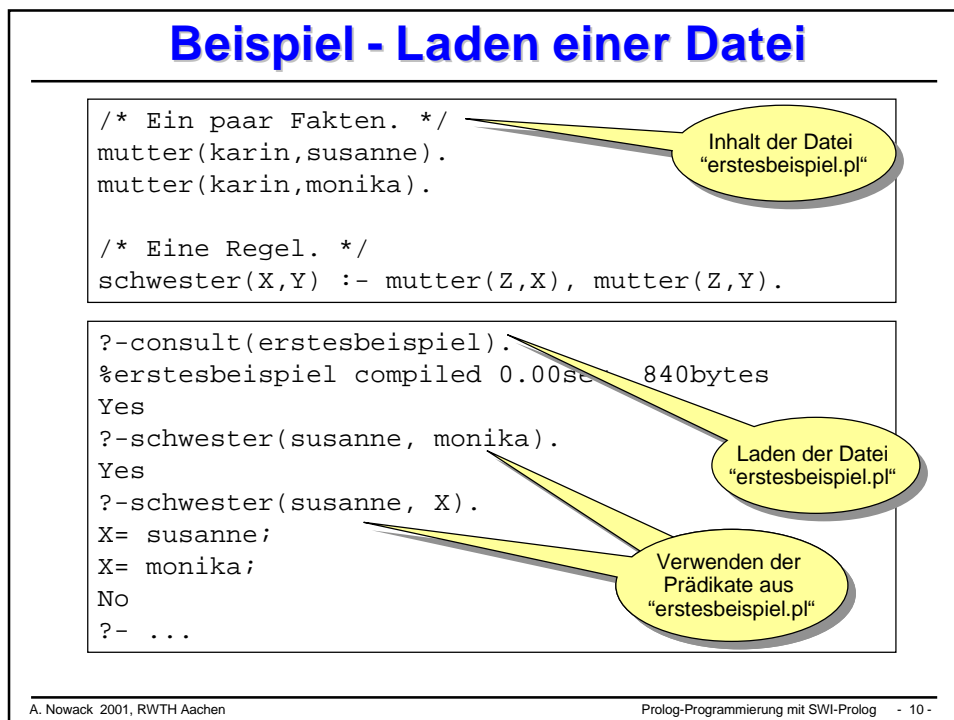
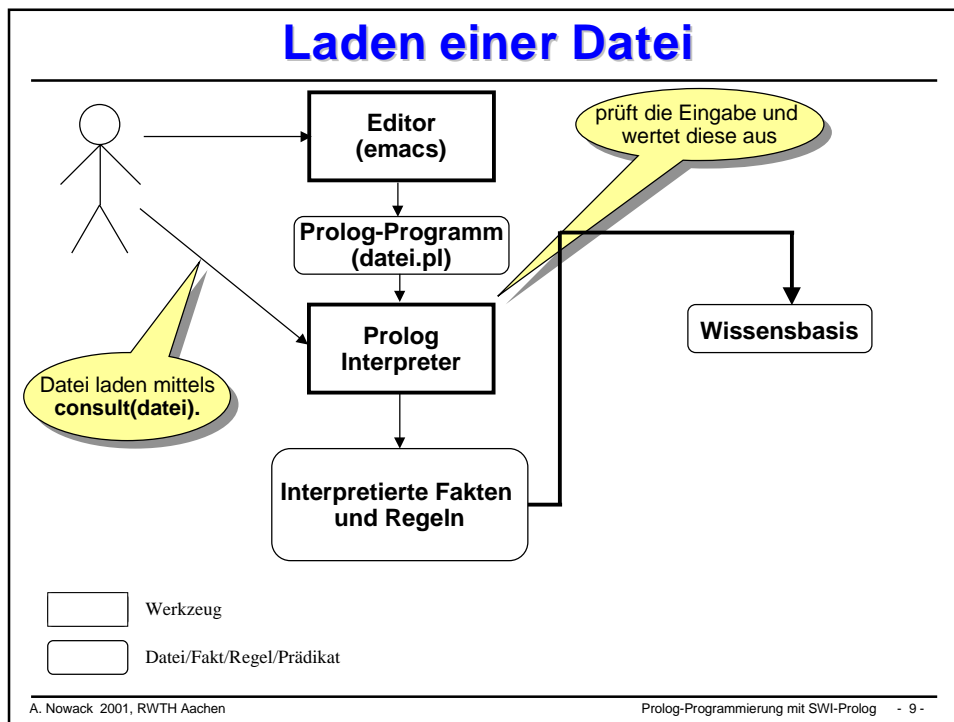
SWI-Prolog starten

- **Installieren von SWI-Prolog (unter Windows)**
 - z.B. von der Erstsemester-CD
 - ◆ selbstentpackendes Archiv
 - ◆ den Dialogen folgen
- **Starten der SWI-Prolog-Umgebung**
 - z.B. über den entsprechenden Menü-Eintrag im Programmverzeichnis
- **Danach Eingabe von Anfragen möglich:**

```

...
?- <( 3 , 2 ) .
No
?- ...
          
```

A. Nowack 2001, RWTH Aachen
Prolog-Programmierung mit SWI-Prolog - 8 -



Anmerkungen

- Jede Klausel (d.h. Fakt, Regel, Anfrage) wird mit einem **Punkt** abgeschlossen.

- **Kommentare** beginnen mit /* und enden mit */

- Laden einer Datei unter Windows ist auch direkt über den Windows-Explorer möglich („Doppelklick“).
 - Das **Verzeichnis**, auf welches der Interpreter dann beim Laden weiterer Dateien mit Hilfe von consult zugreift, ist das der geladenen Datei.
 - Sonst ist dies das voreingestellte Verzeichnis.

Fehler (1)

- Bei einem syntaktischen Fehler wird die **Ausführung abgebrochen** und eine entsprechende Fehlermeldung produziert.

- Es können jedoch nicht nur syntaktische Fehler auftreten sondern auch Fehler folgender Art:
 - eine (gemäß der Aufgabenstellung) fehlerhafte Antwort
 - unendlicher Beweis

- Bei solchen (nicht syntaktischen) Fehlern können **vordefinierte Prädikate** die Fehlersuche unterstützen:
 - **trace**
 - ◆ zur Ausgabe des Beweisverlaufes
 - ◆ beim Beweis des Literals **notrace**
 - wird als Nebeneffekt der trace-Modus ausgeschaltet
 - ◆ wird hier genauer erläutert

Fehler (2)

- **spy**
 - ◆ setzen von Beobachtungspunkten auf vom Benutzer definierte Prädikate (das Prädikat muß übergeben werden)
 - ◆ beim Beweis des Literals **nospyp**
 - werden als Nebeneffekt die Beobachtungspunkte von dem in P spezifizierten Literal entfernt

- **debugging**
 - ◆ Ausgabe der gerade gesetzten Beobachtungspunkte
 - ◆ beim Beweis des Literals **nodebug** werden als Nebeneffekt
 - alle gesetzten Beobachtungspunkte gelöscht
 - der debug-Modus ausgeschaltet

Fehleranalyse mit trace

- Beim Beweis des Literals **trace** wird als Nebeneffekt der trace-Modus eingeschaltet.

- Im **trace-Modus**
 - wird jeder einzelne Beweisschritt des Prologsystems ausgegeben
 - danach auf eine Reaktion des Benutzers gewartet

- Beim eventuellen Backtracking wird der trace-Modus nicht ausgeschaltet.

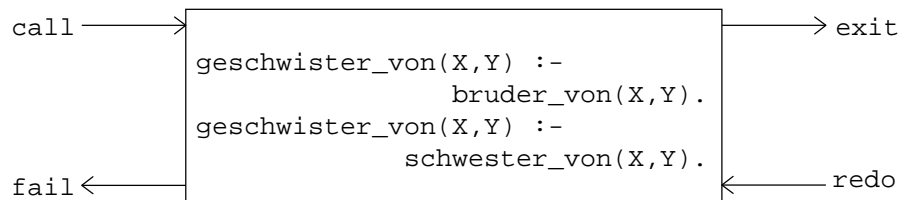
- Beim Beweis des Literals **notrace** wird als Nebeneffekt der trace-Modus ausgeschaltet

Ausgabe des Beweisverlaufs

- Bei der Ausgabe des Beweisverlaufs liegt das sogenannte **Box-Modell** zugrunde.
- Wenn ein Literal bewiesen werden soll, dann wird dieses **Literal als Box** dargestellt.
 - zwei **Eingänge**: call und redo
 - zwei **Ausgänge**: exit und fail
 - in der Mitte: die in der Datenbank vorhandenen Klauseln für das entsprechende Literal

Beispiel für das Box-Modell

- Alle Klauseln des Beipiels der Verwandtschaftsbeziehungen seien bekannt.
- Zu beweisendes Literal:
`geschwister_von(hanna, wilhelm)`
- zugehöriges Box-Modell:



Bedeutung der Pfeile

■ call

- **erster Beweisversuch** des Literals

■ exit

- **erfolgreicher Beweis** des Literals

■ redo

- durch diesen Eingang betritt das Prolog-System die Box, wenn es **schon einen Beweis für das Literal gefunden** hat, aber ein **nachfolgendes Literal nicht bewiesen** werden konnte.
- Ein **neuer Beweis** für das Literal muß gesucht werden.

■ fail

- **Verlassen** der Box, wenn
 - ◆ **keine** in der Box stehende **Klausel zum Beweis** des Literals **verwendet werden kann** oder
 - ◆ **bei allen verschiedenen Beweisen** des Literals ein **nachfolgendes Literal nicht bewiesen werden konnte**

A. Nowack 2001, RWTH Aachen

Prolog-Programmierung mit SWI-Prolog - 17 -

Steuerung des trace-Modus

■ creep

- nächstes Betreten oder Verlassen einer Box soll ausgegeben werden
- Benutzer wird erneut aufgefordert, den trace-Modus zu steuern

■ skip

- Es soll erst wieder eine Ausgabe erfolgen, wenn die gerade aktuelle Box betreten oder verlassen wird.

■ leap

- Es soll erst wieder eine Ausgabe erfolgen, wenn ein Beobachtungspunkt erreicht wird oder wenn die aktuelle Box betreten oder verlassen wird.

■ abort

- Der Beweis soll abgebrochen werden.

A. Nowack 2001, RWTH Aachen

Prolog-Programmierung mit SWI-Prolog - 18 -

Zusammenfassung

■ Prolog

- ist ein Beispiel einer logischen, deklarativen Programmiersprache
- beantwortet Anfragen unter Zuhilfenahme einer Wissensbasis

■ SWI-Prolog

- frei verfügbar und unter GNU-Lizenz verteilbar
- ist auch auf der Erstsemester-CD

■ Arbeit mit SWI-Prolog

- Fakten und Regeln werden aus einer Datei geladen
- Anfragen werden direkt eingegeben

■ Fehler

- Ausgabe des Beweisverlaufes mit Hilfe des Literals trace ermöglicht Analyse