

## Software-Technologie

- Software-Systeme sind sehr komplex. Im Idealfall erfolgt die Programmierung problemorientiert, während die notwendige Übertragung in ausführbare Programme automatisch erfolgt.
- Prolog-Philosophie: Program = Logic + Control .  
Dabei übernimmt der Interpreter den „Control“-Teil.

Fragen zum Nachdenken: Was braucht man für eine gut nutzbare Software-Technologie? Wie können viele Programmierer an einem Software-Projekt arbeiten?

## Prolog basiert auf Prädikatenlogik

- PK1: Exakter, theoretisch beherrschter Kalkül.
- Eine Theorie wird durch eine Menge gültiger Sätze (Theoreme) beschrieben. Sie kann semantisch (inhaltlich) oder syntaktisch (formale Regeln) definiert werden.
- Die Logik ist die Basis für die Beschreibung von Theorien.
- Aus einer Axiomenmenge können Sätze abgeleitet (formales Beweisen) oder gefolgert (Gültigkeit) werden.

Fragen zum Nachdenken: Wie lassen sich Ableiten und Folgern definieren? Wodurch unterscheiden sie sich?

## Beweise

- Beweise können inhaltlich (Argumente) oder rein formal (Umformung von Zeichenketten) erfolgen.
- Formale Beweise (Manipulation von Zeichenketten) können von Maschinen ausgeführt werden (Theorembeweiser).  
Zumindest theoretisch kann man damit beliebige Probleme lösen.

Fragen zum Nachdenken: Wo liegen in der Praxis die Grenzen für den Einsatz von Theorembeweisern? Ist der Prolog-Interpreter ein Theorembeweiser?

## Vollständigkeit, Korrektheit etc.

- Der Vergleich unterschiedlicher Ansätze (Folgern, Ableiten) bzw. unterschiedlicher Beschreibungen von Theorien führt wieder auf Fragen der Vollständigkeit bzw. Korrektheit.
- Die Formalisierung von Theorien führt auf Probleme der Entscheidbarkeit bzw. Aufzählbarkeit (Axiomatisierbarkeit).
- Hinsichtlich der Realität bzw. Nutzbarkeit muss nach der Adäquatheit von Beschreibungen gefragt werden.

Fragen zum Nachdenken: Inwiefern stimmen Ableiten und Folgern im PK1 überein? Was bedeutet Monotonie?

## Kalküle

- Beweisverfahren lassen sich als Suchverfahren betrachten. Dabei ist die Beweisrichtung wesentlich:
  - Vorwärts: Deduktiver Kalkül
  - Rückwärts: Testkalkül
- Negative Kalküle (Widerspruchsbeweis) sind oft effizienter.

Fragen zum Nachdenken: Warum sind Negative Kalküle effizienter?

## Klauselform

- Eine Klausel ist eine Menge von Literalen.
- Ein Literal ist eine negierte oder nicht negierte Atomformel (Prädikat).
- Logische Ausdrücke können über mehrere Schritte in erfüllbarkeitstheoretisch äquivalente Mengen von Klauseln umgeformt werden.

Fragen zum Nachdenken: Warum ist die Notation in der Schreibweise von Klauseln vorteilhaft?

## Resolution

- Allgemeingültigkeitsbeweise (bzw. Unerfüllbarkeitsbeweise) können im Resolutions-Kalkül geführt werden. Dabei wird versucht, die leere Klausel (einen Widerspruch) abzuleiten.
- Die ursprüngliche Formel ist genau dann allgemeingültig, wenn aus der nach Umformungen entstehenden Klauselmengende die leere Klausel abgeleitet werden kann.

Fragen zum Nachdenken:

Man kann mithilfe des Resolutions-Verfahrens auch testen, ob eine logische Formel aus einer Menge von Formeln (Axiomen) folgt. Wie geht das?

Der PK1 ist nicht entscheidbar. Besteht hier ein Widerspruch?

## Resolution

- Bei der Resolution werden aus Klauseln neue Klauseln erzeugt. Dazu müssen Mengen von Literalen durch Unifikation syntaktisch identisch gemacht werden.
- Inhaltlich bedeutet das die gleichzeitige Ausführung von Resolutionsschritten für unterschiedliche Variablenbelegungen.
- Ein allgemeinsten Unifikator kann konstruiert werden.

Fragen zum Nachdenken: Wann ist eine Substitution spezieller als eine andere? Um was für eine Relation handelt es sich bei „spezieller als“?

## Resolutionsstrategien

- Unterschiedliche Auswahl- und Reihenfolgekriterien führen zu unterschiedlichen Resolutionsstrategien. Es entsteht wieder die Frage nach Vollständigkeit bzw. Korrektheit der Verfahren (Suchverfahren nach der leeren Klausel).

Fragen zum Nachdenken:

Die Resolutionsmethode ist nicht vollständig (im Sinne des Ableitens), aber widerlegungsvollständig. Warum ist das ein Vorteil?

## Prolog-Interpreter

- Der Prolog-Interpreter realisiert eine SLD-Resolution.
- Prolog-Programme sind auf HORN-Klauseln beschränkt.
- Damit ist die deklarative Semantik von „reinem“ Prolog festgelegt.
- Als Programmiersprache besitzt Prolog weitere Möglichkeiten, die die Semantik verändern.
- Das Laufzeitsystem des Prolog-Interpreters sorgt für eine effiziente Verarbeitung.

Fragen zum Nachdenken: Ist die SLD-Resolution vollständig und korrekt?