1. Introduction

Why different prog. languages?
- Choose the right language for the right application
- Eases learning of new prog. languages

Main application area of logic programming:
- artificial intelligence
- expert systems
- robotics

Several Prolog implementations:
http://www.swi-prolog.org

Ex: Family Data Base as a Logic Program

Prolog = Programming in Logic
Prolog-Reg. consists of facts and rules.

Fact: predicate symbol (obj1, ..., objn).

Start with lower-case symbols

Comments in Prolog:

% ... end of line

or
/

?/
Execution of logic prog: ask queries

?- male(gerd).
true

?- married(monika, werner).
false

^ Prolog uses a closed-world- assumption:
If a statement is not implied by the formulas in the program, then it must be false.

To execute a program, one has to "load" it.

\[ \text{Consult} \]

\underline{Variables in Programs}

Extend prog. by the following fact:

human(X).

^ variables start with capital letters

Variables in a program are universally quantified (stand for all possible instantiations)

?- human(gerd).  ?- human(5).
true        true
If a variable occurs several times in the same clause, then all of its occurrences have to be instantiated in the same way.

\[ \text{likes}(X, Y). \quad \text{everybody likes everybody} \]

\[ \text{likes}(X, X). \quad \text{everybody just likes him/herself} \]

Variables in Queries

\[ ? - \text{motherOf}(X, \text{susanne}). \]

Variables in queries are existentially quantified.

"Who is the mother of susanne?"

\[ X = \text{renate} \]

\[ ? - \text{motherOf}(\text{renate}, Y). \quad \text{"Who are the children of renate?"} \]

\[ Y = \text{susanne}; \]

\[ Y = \text{peter} \quad \text{stands for "or"} \]

Can be used to let Prolog continue searching for solutions.

In Prolog, the query determines what is input/output.
In Prolog, the query determines what is input/output.

**Combination of Queries**


"Is gerd the father of susanne?"

Prolog first solves `married(gerd, W)`

\[ W = \text{renate} \]

Then it solves `mother_of(renate, susanne)`.

- Prolog treats queries from left to right.
- \[ \text{Clauses in a prog. from top to bottom} \]

```prolog
?- mO(G, M), mO(M, aline)
\[ G = \text{monika} \]
\[ M = \text{karin} \]
\[ M = \text{klaus} \]
\[ G = \text{renate} \]
\[ M = \text{susanne} \]
\[ mO(\text{karin}, aline) \]
\[ mO(\text{klaus}, aline) \]
\[ mO(\text{susanne}, aline) \]
\[ \triangleright \]
```

Prolog constructs a proof tree and backtracks in case of failure.

It stops as soon as \[ \triangleright \] is reached and returns the instantiation of the variables that corresponds to the path from the root to \[ \triangleright \].

**Rules**
Rules are needed to deduce new knowledge from existing knowledge.

\[
\text{fatherOf}(F, C) :- \text{married}(F, W), \text{motherOf}(W, C).
\]

\[\text{head} \quad \text{if} \quad \text{body}\]

\[
\text{fatherOf}(\text{gerd}, Y) \\
\text{married}(\text{gerd}, W), \text{motherOf}(W, Y) \\
W = \text{renate} \\
\text{motherOf}(\text{renate}, Y) \\
Y = \text{susanne} \quad \text{or} \quad Y = \text{peter}
\]

Several rules for the same predicate

\[
\text{parent}(X, Y) :- \text{motherOf}(X, Y).
\]
\[
\text{parent}(X, Y) :- \text{fatherOf}(X, Y).
\]

\[
\text{Alternative:} \\
\text{parent}(X, Y) :- \text{motherOf}(X, Y) \land \text{fatherOf}(X, Y).
\]

Recursive Rules

Recursion can lead to non-termination.
Characteristics of Logic Programming:

- Programs are just a collection of facts and rules, no control structures.
- LP resulted from automated theorem proving, program does not determine input/output, but this depends on the query.
- LP particularly suitable for AI, deductive databases, rapid prototyping.

Contents of the Lecture:

1. Intro to LP
2. Basics of Predicate Logic
3. Resolution (Proof Principle used in LP)
4. Syntax + Semantics of Pure Logic Programs
5. The Programming Language Prolog
6. Constraint Logic Programming
Organization

- english
- german course notes (web)
- english handwritten course notes & notes from the lecture, also on the web
- video (2013)
- web site: http://verify.rwth-aachen.de/lp17
  includes info on references, software, news, transparencies, exercise sheets, ...

- weekly exercise sheets
  - solve in groups of 3
  - first ex. sheet this Friday (April 28)
  - return ex. sheets next Friday
    - beginning of ex. course or
      - wooden box, EN, 2nd floor
  - 50% of the points needed to participate in the exam

- register to participate in the exercises until Friday (April 28)
  ON OUR WEBSITE
  (no L2P, no reg. for exercises on...