1. Introduction

Freitag, 10. April 2015 08:30

Logic Programming

Learning several prog. languages:

- express ideas during SW development
- needed to decide which language to use in project
- eases learning of new languages
- needed to design new languages

Imper. + Fct. Languages:

programs compute functions

Logic Languages:

- programs describe relations
- execution: ask queries, program tries to prove queries
- main application area: AI, expert systems, deductive data bases, ...

Prolog-Implementation: SWI Prolog (see web page)

Example: Family Tree

— : married

↙ : children

Facts and Queries

Knowledge has to be translated to Prolog-Syntax.

Prolog = Programming in Logic

Prolog program consists of (special) logic formulas, so-called <u>clauses</u>:

- facts
- rules (allow to deduce new knowledge from existing knowledge)

Syntax of facts:

predicate (obj1, ..., objn). ← Boolean statements
symbol
↑
Strings starting with lower-case letter

Relations are not symmetric.

Syntax ~~for~~ comments:

% ... end of line or

```
/*
  ⋮
*/
```

Execution: ask queries

?- statement.

Closed World Assumption:
everything that can't be deduced from prog. clauses is <u>false</u>

Variables in Programs

Variables: strings starting with capital letter or with __

Variables in programs are <u>universally</u> quantified

Ex: all X are human
(i.e., everything is human)

Same variables in one clause have to be instantiated in the same way:

likes(X,X). - everybody likes himself

likes(X,Y). - everybody likes everybody

Variables in Queries

Variables in queries are existentially quantified — can be used to let the program compute solutions.

Ex: Who is the mother of susanne?
(Is there an X such that ... ?)
Prolog returns a suitable answer substitution.

If there are several solutions: ; makes Prolog continue searching for answers.

Prolog searches through its prog. clauses from top to bottom.

Same program can be used to compute mothers or children ⟹

Prolog programs have no fixed input/output, but input/output depends on query.

```
?- motherOf(X,Y).
X = monika, Y = Karin ;
  ⋮
```

```
?- human(Z).
true
```

← Prolog returns the most general instantiations that make the query true.

Combined Queries

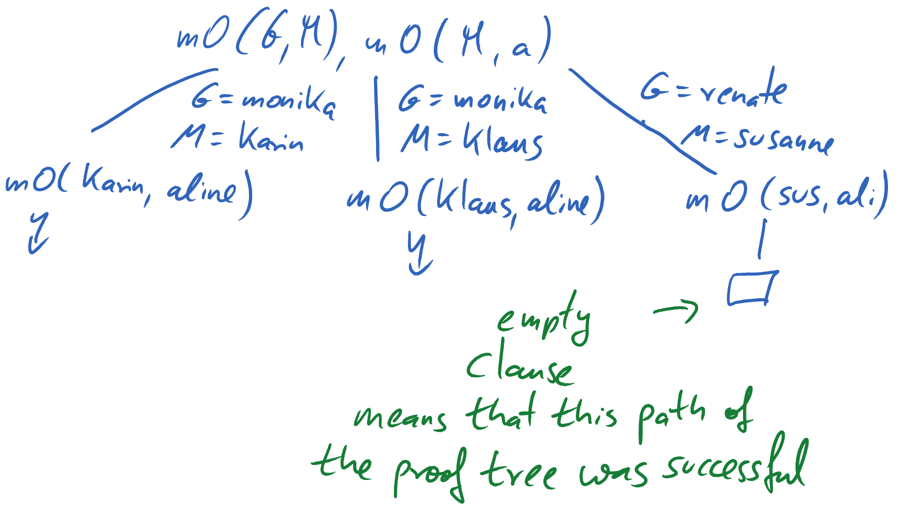
Combined Queries

, ≙ and ; ≙ or

Ex: Is gerd the father of susanne?

Combined queries are executed from *left to right*

- first solve query married(gerd, W)
  ⇒ finds an instantiation of W
- then solve second query
  motherOf(W, susanne) for this instantiation of W
- If second query fails, then backtrack to the first query and try the next solution.
- Prolog computes a proof tree (so-called SLD tree)



Rules

rules allow to deduce new knowledge from existing knowledge

Ex: F is the father of C if (:-)
there exists a W such that

F is married to W and
W is the mother of C.

Rules: $\text{head} \;\text{:-}\; \underbrace{\text{statement}_1, \dots, \text{statement}_n.}_{\text{body of the rule}}$

means: in order to prove head,
one can instead prove the statements
in the body

Ex: fO(gerd, Y)

| F = gerd, C = Y

married(gerd, W), motherOf(W, Y)

| W = renate

motherOf(renate, Y)

/ Y = susanne   \ Y = peter

□   □

Several rules for the same predicate

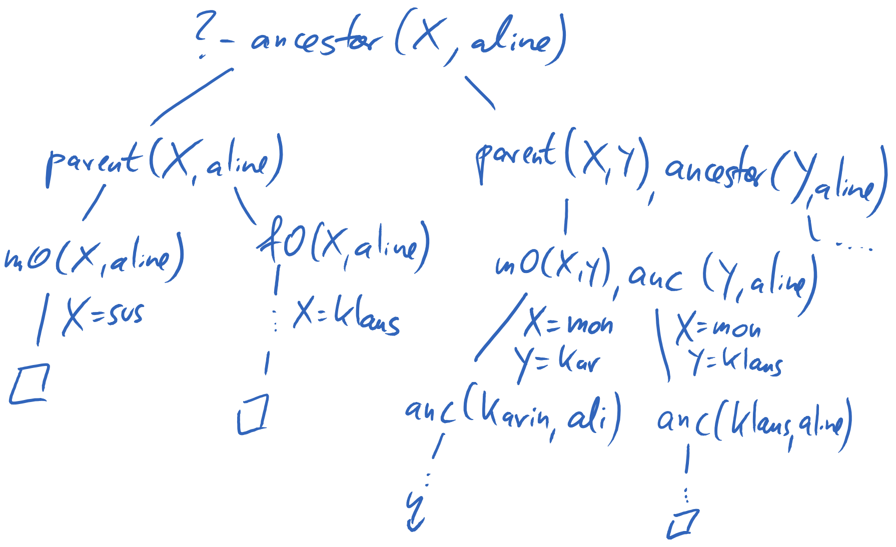
alternative:
parent (X, Y) :- motherOf (X, Y) ; fatherOf(X, Y).

(; is defined by 2 clauses)

?- parent (X, susanne).
Mother will be found first due to the order
of prog. clauses.

Recursive Rules

Ex: ancestor predicate
2nd rule is recursive



Characteristics of Logic Programming:

- no control structures, just facts + rules
- prog. execution ≙ automated theorem proving
- particularly suitable for AI

Plan for the lecture:

Ch. 1: Introduction to LP
Ch 2: Predicate Logic
Ch 3: Resolution (Proof Technique used in LP)
Ch 4: Syntax + Semantics of LP
Ch 5: Prog. Language Prolog

Organisation

- english
- german course notes (web)
- english notes from the lecture + Slides (web)
- lecture: 8:30 – 10:00 mon + fri
  exercise: 10:15 – 11:45 fri
- video recording from 2013

- V3+Ü2 lecture, 2 variants (for Bachelor + Master Students) called V3B + V3M (Math students: V3B)
- Web site: http:/ verify.rwth-aachen.de / lp15
- Exercises:
  - weekly exercise sheet
  - groups of 2
  - registering for exercises: via our web site (until Friday next week)
  - 50% of exercise points needed to participate in the exam
  - exam: August 19 + September 14
- Vorgezogene Masterprüfung: register via ZPA (June 8 - 18)