Universität Freiburg
Institut für Informatik

Dr. Fang Wei

Georges-Köhler Allee, Geb. 51
D-79110 Freiburg

fwei@informatik.uni-freiburg.de

Foundations of Query Languages
Summerterm 11
Discussion by 27.07.2011

5. Datalog

Exercise 1 (Datalog)
Encode words over the alphabet $\{a, b\}$ structures having the following relations:

- $Min(X)$: expressing that X is the first position of the word.
- $Max(X)$: expressing that X is the last position of the word.
- $Succ(X, Y)$: expressing that the position Y is the successor position of X .
- $P_a(X)$: position X contains letter a.
- $P_b(X)$: position X contains letter b.

a) Write a datalog program that makes an atom *yes* true iff there are more a's than b's in the string.
b) Write a datalog program that makes an atom *yes* true iff the word is a palindrome.

Exercise 2 (Datalog)
Give the well-founded semantics for the following Datalog programs:

a) $p \leftarrow \neg q$
$q \leftarrow \neg r$
$r \leftarrow \neg s$
$s \leftarrow \neg p$

b) $s \leftarrow \neg r$
$p \leftarrow \neg q$
$q \leftarrow p, \neg r$

c) $p \leftarrow q$
$p \leftarrow \neg q$

d) $win(X) \leftarrow move(X, Y), \neg win(Y)$
with EDBs:
$\{move(1, 2), move(2, 3), move(3, 1), move(3, 4)\}$
$\{move(1, 2), move(2, 3), move(3, 1), move(3, 4), move(4, 5)\}$

Exercise 3 (Datalog boundedness)
Consider the following constant-free Datalog program:

$P(x) \leftarrow P_0(x)$
$P(x) \leftarrow R(x, y), P(y)$
$R(x, y) \leftarrow S(x), S(y)$
Is the program bounded? If so, prove your claim and give an equivalent non-recursive Datalog program, otherwise give a counterexample.

Exercise 4 (Propositional Logic Programming)
Let I and J be two models of a propositional logic program P. Prove that the intersection of I and J is also a model of P.

Exercise 5 (P-completeness)
A monotone Boolean circuit contains AND, OR, and $INPUT$ gates, and a single $OUTPUT$ gate. An input for a Boolean circuit consists of an assignment of truth values ($true$ or $false$) to each input gate of the circuit. The determination of the output gate value is known as the Circuit Value Problem (CVP). Prove that the CVP is P-Complete under logspace transformations.