Universität Freiburg
Institut für Informatik

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

Fang Wei-Kleiner

fwei@informatik.uni-freiburg.de

Advanced Databases and Information Systems
Summerterm 2019
Discussion on 18/07/2019

11. Sheet: Datalog

Exercise 1 (Datalog)
We are given two directed graphs G_{black} and G_{white} represented as binary relations over the same set of vertices V. Write a Datalog program that computes the set of $pairs(a, b)$ of vertices such that there exists a path from a to b where *black* and *white* edges alternate, starting with a *white* edge.

$endwithwhite(X, Y) \leftarrow G_{white}(X, Y).$

$endwithblack(X, Y) \leftarrow endwithwhite(X, Z), G_{black}(Z, Y).$

$endwithwhite(X, Y) \leftarrow endwithblack(X, Z), G_{white}(Z, Y).$

$pairs(X, Y) \leftarrow endwithblack(X, Y).$

$pairs(X, Y) \leftarrow endwithwhite(X, Y).$

Exercise 2 (Datalog)
Given are the following three Datalog programs to calculate the transitive closure of a graph:

right-recursive:

$C(X, Y) \leftarrow E(X, Y).\ C(X, Y) \leftarrow E(X, Z), C(Z, Y)$

left-recursive:

$C(X, Y) \leftarrow E(X, Y).\ C(X, Y) \leftarrow C(X, Z), E(Z, Y)$

double-recursive:

$C(X, Y) \leftarrow E(X, Y).\ C(X, Y) \leftarrow C(X, Z), C(Z, Y)$

Use the semi-naive algorithm to calculate the results of the given three programs with respect to the following database:

$E(1, 2), E(2, 3), E(3, 4), E(4, 5), E(5, 6), E(6, 7)$

Exercise 3 (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 .
- $Pa(X)$: position X contains letter a.

- $Pb(X)$: 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.
$S(X,X) \leftarrow Min(X), Max(X), Pa(X)..$ // string is a
$S(X,Y) \leftarrow Min(X), Max(Y), S(X,Z), Succ(Z,Y), Pa(Y)..$ // string is Sa
$S(X,Y) \leftarrow Min(X), Max(Y), S(Z,Y), Succ(X,Z), Pa(X)..$ // string is aS
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z2,Y), S(Z1,Z2)Pa(X), Pb(Y).$ // string is aSb
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z2,Y), S(Z1,Z2)Pb(X), Pa(Y).$ // string is bSa
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z1,Z2), Succ(Z2,Y), S(Z2,Y)Pa(X), Pb(Z1).$ // string is abS
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z1,Z2), Succ(Z2,Y), S(Z2,Y)Pb(X), Pa(Z1).$ // string is baS
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z1,Z2), Succ(Z2,Y), S(X,Z1)Pa(Z2), Pb(Y).$ // string is Sab
$S(X,Y) \leftarrow Min(X), Max(Y), Succ(X,Z1), Succ(Z1,Z2), Succ(Z2,Y), S(X,Z1)Pb(Z2), Pa(Y).$ // string is Sba
$S(X,Y) \leftarrow Min(X), Max(Y), S(X,Z1), S(Z2,Y), Succ(Z1,Z2).$ // string is SS
$true \leftarrow S(X,Y).$

b) Write a Datalog program that makes an atom *yes* true iff the word is a palindrome.
a, b, aa, bb, aPa, bPb