

Implementing a graph algebra

Source: <http://coding.derkeiler.com/Archive/General/comp.theory/2005-11/msg00045.html>

- *From:* "Marius Amado Alves" <amado.alves@xxxxxxxxxx>
 - *Date:* 7 Nov 2005 04:51:34 -0800
-

Dear computer theorists:

I am implementing a graph algebra designed to serve as the query plan component of a graph-based database system. I'm looking for ways to optimize certain cases explained below.

The algebraic type is the set of vertices. The operations include the set theoretic operations and an adjacency operator $A(X)$ with the obvious semantics:

$A(X) = \{y : x \text{ in } X, \{x, y\} \text{ is an edge in the graph}\}$

I have been able to optimize the resolution of expressions like $(\wedge = \text{intersection})$

$A(K)$
 $A(K1) \wedge \dots \wedge A(Kn)$

where K_i is a known set, usually the singleton of a known vertex. I'm looking for ways to optimize expressions of higher order e.g.

$A(A(K))$
 $A(A(K1)) \wedge \dots$

I'm particularly interested in the intersection operation, as it shows up a lot in expressions derived from concrete cases of data modelling and querying (Biopathways, XMark, XQuery Use Cases).

/*

The description above is a slight simplification. The real case is a directed graph, so we have source and target adjacency S, T instead of just A , and hence expressions like

$S(K1) \wedge T(K2) \wedge \dots$
 $S(S(K))$
 $S(T(K))$

Naturally $A(X) = S(X) \cup T(Y)$. S, T are sometimes notated $A+, A-$ in the literature (or the other way around, and that's why I don't use the $+/-$

Implementing a graph algebra

notation, lack of obvious metaphor).

Also, my system has vertex order, which is used in the algebraic sets to optimize certain operations namely \wedge .

*/

Thanks a lot,
Marius A. Alves
PhD student, University of Porto

.

-
- Prev by Date: [*Re: polygons question*](#)
 - Next by Date: [*Re: polygons question*](#)
 - Previous by thread: [*polygons question*](#)
 - Next by thread: [*X3SAT Question*](#)
 - Index(es):
 - ◆ [*Date*](#)
 - ◆ [*Thread*](#)