A Functional and Successor List Based Version of Warshall's Algorithm

Rudolf Berghammer

Institut für Informatik
Christian-Albrechts-Universität Kiel
Olshausenstraße 40, D-24098 Kiel

The computation of the transitive closure $R^+ = \bigcup_{n>0} R^n$ of a homogeneous (binary) relation R on a set X has many practical applications. This is mainly due to the fact that, if R is the set of edges of a directed graph $g = (X, R)$, then R^+ relates two vertices $x, y \in X$ of g if and only if y is reachable from x via a non-empty path. Usually, the task of computing R^+ is solved by Warshall's algorithm [2]. Its traditional implementation in an imperative programming language is based on a representation of the relation R by a 2-dimensional Boolean array. This leads to a simple and efficient program with three nested loops that needs $O(n^3)$ steps, where n is the cardinality of the carrier set X of R.

However, in certain cases arrays are unfit for representing relations. In particular this holds if both R and R^+ are of "medium density" or even sparse. Here a representation of relations by lists or vectors of successor lists is much more economic w.r.t. required space. But such a representation derogates the simplicity and efficiency of the implementation of Warshall's algorithm. Arrays are also problematic if another programming paradigm is used. Especially, the method of imperatively updating an array representing the relation is alien to the purely functional programming paradigm which restricts the use of side effects.

In the talk we show how systematically to obtain a purely functional version of Warshall's algorithm that bases on an implementation of relations by linear lists of successor lists and also has cubic running time. Using relation algebra in the sense of [1] as methodical tool, in the first step we develop a functional algorithm for computing R^+ that solely bases on the operations of relation algebra and a choice operation that selects a point contained in a non-empty vector. To obtain from it a version that works on lists of successor sets, we represent in the second step relations on X by functions from X into its powerset (which map elements to its successor sets) and vectors on X by elements of 2^X (viz. the set they describe) and, after that, in the third step the functions by lists over 2^X. Going, finally, from elements of 2^X to lists over X and from lists over 2^X to lists of lists over X, we obtain a version that immediately can be translated into the functional programming language Haskell.

Literatur

1. Schmidt G., Ströhlein T.: Relations and graphs. Discrete Mathematics for Computer Scientists, Springer (1993).
2. Warshall S.: A theorem on Boolean matrices. Journal of the ACM 9, 11-12 (1962).