Query Languages for Sequence Databases: Termination and Complexity*

Giansalvatore Mecca
Università della Basilicata (Italy)
mecca@dia.uniroma3.it

Anthony J. Bonner
University of Toronto (Canada)
bonner@cs.toronto.edu

Abstract

This paper develops a query language for sequence databases, such as genome databases and text databases. Unlike relational data, queries over sequential data can easily produce infinite answer sets, since the universe of sequences is infinite, even for a finite alphabet. The challenge is to develop query languages that are both highly expressive and finite. This paper develops such a language as a subset of a logic for string databases called *Sequence Datalog*. The main idea is to use safe recursion to control and limit unsafe recursion. The main results are the definition of a finite form of recursion, called *domain–bounded recursion*, and a characterization of its complexity and expressive power. Although finite, the resulting class of programs is highly expressive, since its data complexity is complete for the elementary functions.

1 Introduction

It is widely accepted that relational databases do not provide enough support for many of today's advanced applications. In some cases, object-oriented databases are the right solution. However, in other cases, such as genome databases and text databases, there is still a need for more flexibility in data representation and manipulation. The problem of extending relational databases with string manipulation features has recently motivated several research proposals [4, 8, 5, 6, 7]. In fact, sequences represent a particularly interesting domain for query languages. In contrast to sets, computations over sequences can easily become infinite, even when the underlying alphabet is finite. This is because repetitions of symbols are allowed, so that the number of possible sequences over any finite alphabet is infinite. The researcher thus faces an interesting challenge: on the one hand, the language should provide powerful primitives for restructuring sequences; on the other hand, the expressive power of the language should be carefully limited, to avoid infinite computations.

In [3], we developed a logic called *Sequence Datalog* for querying sequence databases. A safe subset of the logic was defined, based on a new computational model called *Generalized Sequence Transducers*. These machines are a simple yet powerful device for computing sequence mappings. In [3], we showed how networks of these machines could be expressed in Sequence Datalog. Moreover, any Sequence Datalog program constructed in this way is guaranteed to be safe and finite. In this paper, we take a different approach: instead of computational definitions, we develop *syntactic restrictions* that guarantee finiteness and safety. This provides an alternate view of finite computations in the logic. The main idea is to use structural recursion (which is guaranteed to terminate) to limit the construction of new sequences. The

*Preliminary portions of this paper appear in the *Proceedings of the Fifth International Workshop on Database Programming Languages (DBPL'95).* The first author was partially supported by MURST and Consiglio Nazionale delle Ricerche (CNR). The second author was partially supported by an operating grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).