Queries and Computation on the Web*

Serge Abiteboul1 ** and Victor Vianu2

1 Computer Science Department, Stanford University, Stanford, CA 94305-9045
2 Univ. of California at San Diego, CSE 0114, La Jolla, CA 92093-0114

Abstract. The paper introduces a model of the Web as an infinite, semi-structured set of objects. We reconsider the classical notions of genericity and computability of queries in this new context and relate them to styles of computation prevalent on the Web, based on browsing and searching. We revisit several well-known declarative query languages (first-order logic, Datalog, and Datalog with negation) and consider their computational characteristics in terms the notions introduced in this paper. In particular, we are interested in languages or fragments thereof which can be implemented by browsing, or by browsing and searching combined. Surprisingly, stratified and well-founded semantics for negation turn out to have basic shortcomings in this context, while inflationary semantics emerges as an appealing alternative.

1 Introduction

The World Wide Web [BLCL+94] is a tremendous source of information which can be viewed, in some sense, as a large database. However, the nature of the Web is fundamentally different from traditional databases and raises qualitatively new issues. Its main characteristics are its global nature and the loosely structured information it holds. In this paper, we consider some fundamental aspects of querying the Web.

We use as a model of the Web an abstraction that captures its global nature, and the semi-structured information it holds. Perhaps the most fundamental aspect of our model is that we view the Web as infinite. We believe this captures the intuition that exhaustive exploration of the Web is –or will soon become– prohibitively expensive. The infiniteness assumption can be viewed as a convenient metaphor, much like Turing machines with infinite tapes are useful abstractions of computers with finite (but potentially very large) memory. Note that our approach is fundamentally different from previous attempts to model infinite data (e.g. [HH93, KKR90]) which focus on finitely representable databases. In contrast, we do not assume the Web is finitely represented. Instead, we view it as

* Work supported in part by the National Science Foundation under grant number IRI-9221268.
** This author’s permanent position is INRIA-Rocquencourt, France. His work was supported in part by the Air Force Wright Laboratory Aeronautical Systems Center under ARPA Contract F33615-93-1-1339, and by the Air Force Rome Laboratories under ARPA Contract F30602-95-C-0119
a possibly nonrecursive infinite structure which can never be entirely explored. Our model leads to a focus on querying and computation where exploration of the Web is controlled. This raises issues akin to safety in classical databases.

The data model we use is similar to several models for unstructured data recently introduced, e.g., [Q+95, CACS94, BDS95]. The Web consists of an infinite set of objects. Objects have a value and/or may reference other objects via labeled links. The set of labels for each object is not fixed, unlike the attributes of a relation. Intuitively, an object can be viewed as a Web page; the value is the content of a page; labels provide links that allow navigating through the Web, in hypertext style.

We begin by exploring the notion of computable query in the context of the Web. Our model is along the lines of the computable queries of Chandra and Harel [CH80]. We introduce a machine model of computation on the Web that we call a Web machine. This works much like a Turing machine, but takes as input an infinite string and may produce an infinite answer. We also introduce two particular machine models that capture directly the main styles of computing used on the Web: browsing and searching. The browser machine model allows for navigational exploration of the Web. The browse/search machine additionally allows searching in the style of search engines.

Based on the Web machine, we define the notions of computability and eventual computability of queries. The latter notion arises from the fact that infinite answers to queries are allowed. A query is computable if its answer is always finite and computable by a halting Web machine. A query is eventually computable if there is a Web machine, possibly nonterminating, which eventually outputs each object in the answer to the query. Interesting connections hold with the browser machine and with the browse/search machine. We show that every generic and computable query is in fact computable by a browser machine. This confirms the intuition that browsing is in some sense the only way to control computation on the Web. We also show that the set of generic queries which are eventually computable by a Web machine is precisely the same as the set of generic queries which are eventually computable by a browse/search machine. Thus, everything can be done by a combination of browsing and searching.

To express queries, one needs query languages. We are interested in the ability of declarative database query languages to express queries on the Web. To this end, we revisit the classical languages FO (first-order logic), Datalog, and Datalog~. The questions of interest for each language are the following: (i) Are the queries in the language computable or eventually computable? (ii) Which fragments of each language can be implemented by browsers and which by a combination of browsing and searching? We provide syntactic restrictions that guarantee computability by browsers or by browse/search machines in FO and Datalog~.

One of the interesting results of the paper is with respect to negation. The “positive” fragment of FO is eventually computable. The addition of recursion yields no problem. However, negation brings trouble, and some simple FO queries are not eventually computable. The Datalog~ languages yield some surprises: