Reasoning about Client-side Web Programs

Philippa Gardner
London Imperial College, UK

Abstract

In PODS 2008, I presented a paper on a formal, compositional specification of the Document Object Model (DOM), a W3C XML Update library. This work concentrated on Featherweight DOM, a small fragment of DOM which focuses on the XML tree structure and simple text nodes. Since the formal reasoning is compositional, we are able to work with a minimal set of commands and obtain complete reasoning for straight-line code. We are also able to verify, for example, invariant properties of simple DOM programs.

This work is based on a recent breakthrough in program verification, based on analysing a program's use of resource. The idea is that the reasoning should follow the programmers' intuitions about which part of the computer memory the program touches. This style of reasoning was introduced by O'Hearn (Queen Mary) and Reynolds (CMU) in their work on Separation Logic for reasoning modularly about large C-programs (e.g. Microsoft device driver code, Linux). I substantially extended the range of local resource reasoning, introducing Context Logic to reason about programs that directly manipulate complex data structures such as XML.

In this survey talk, I will give an overview of our theoretical and practical work on reasoning about DOM, highlighting recent developments which include:

1. the extension of this work to DOM Core Level 1. A substantial piece of work, not because of the reasoning, but because full DOM is large, underspecified and difficult to interpret, with no consensus between browsers;
2. reasoning about the combination of JavaScript and DOM to provide, for example, secure mashups for a more flexible, secure integration of outsourced payment services;
3. on-going work on a verification tool for automatically reasoning about DOM programs and the identification of key examples of web applications on which to test our DOM reasoning: e.g., with current technology, we can prove by hand that mashup programs are fault free; with our tool, such proofs will be automatic.

An ultimate challenge is to develop the necessary reasoning technology to provide a safe and secure web environment on which to build the next generation of web applications, thus demonstrating the scientific feasibility of a reliable Web.