Lisp in Summer Projects Submission

Submission Date	2013-10-24 08:20:56
Full Name	Evan Donahue
Country	USA
Project Name	Boris The (Web) Spider
Type of software	library
General category	library
LISP dialect	Racket
GitHub URL	https://github.com/emdonahu/boris
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	Boris aims to make getting data off of a website and into a running program as cheap, reliable, maintainable, verifiable, extensible, and reusable as possible.
Function	Boris defines an embedded language for guiding a web client through a series of web pages, collecting data and returning results to the host program for immediate use.
Motivation	I have seen too many good ideas shelved do to the hassle of managing complex crawls through messy markup. With Boris, I hope to enable myself and others to get up and running quickly with programs that interact dynamically with web content.
Audience	Boris is for programmers who want to write programs that interact with the web without having to track cookies and connections, parse markup, or manage intermediate results; this might include programmers working on web service mashups, text-mining research, or web data visualization.
Methodology	Boris has three core concepts: webs, spiders, and flies. A web is a program structure defining the urls to visit and the data to extract. A spider is a function that interprets the web and performs the crawl, handling concerns external to the web logic such as caching, robots.txt, and timing policies.

The flies are the data extracted by the spider as it traverses the web and returned to the host program (via return value, yield, etc.).

A web is a tree in which each node is a function. The function accepts the current crawl state (containing the current page, environment bindings, and position in the web) and returns a list of "next" states. The spider then executes the subsequent portions of the web for each of these states in turn. In this way, Boris forms can inherently express concepts such as branching (an empty list prunes the sub-tree beneath the current node) and iteration (a full list repeats the sub-tree code for each state). This way, concepts such as "follow every link in the navigation div" have natural expressions in Boris.

All of Boris' basic web forms translate to one of four semantic operations. Navigation forms request new pages from external sources, such as by fetching a document from the web. Extraction forms return data discovered during the crawl to the host program. Binding forms bind identifiers in the spider's memory environment for use later in the crawl, which can be useful for assembling extractable entities whose data is spread across several pages. Control forms dynamically alter the sub-trees the spider will visit next, allowing for control constructs such as recursion, which can concisely express common tasks like pagination and full-site mirroring. Because a web is a simple tree of functions, however, it is possible to add new forms at runtime simply by modifying the tree.

Conclusion

The primary goal of this iteration of Boris is to devise a web spider definition language that is sufficiently expressive to handle a wide range of web crawling tasks easily and efficiently. This goal seems to have been reached. Boris' primary area for growth right now is its development story. Boris is designed to play target language for an eventual textual (and possibly even graphical) browser-like interface capable of turning exploratory web browsing directly into runnable spidering code, while still preserving the ability to escape to Boris' more complete programming model when the affordances of the browser fail.

Build Instructions

Have Racket 5.9 installed.

Run on the command line:
```
raco pkg install github://github.com/emdonahu/boris/master
```

Test Instructions

Unit tests can be run with:
```
raco test -px boris
```

More involved demos can be found in the tests/boris folder, and run with:
```
racket
```

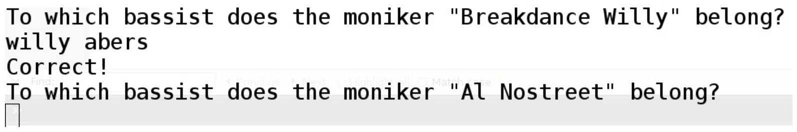
Execution Instructions

Being a library, the execution instructions are the same as the test instructions. Additional documentation can be found at http://emdonahu.github.io/boris/index.html

Describe any bugs or caveats

Some of the tests are network tests, and so depend on port 60415 being clear for a local web server to run.

Screen shots



```
To which bassist does the moniker "Breakdance Willy" belong?
willy abers
Correct!
To which bassist does the moniker "Al Nostreet" belong?
```

boris2.png

borisscreen.png

Official

I have read rules and have abided by them.
I am 18 years of age or older.
I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.