JSCert: One Year On

Philippa Gardner Gareth Smith
Conrad Watt Thomas Wood

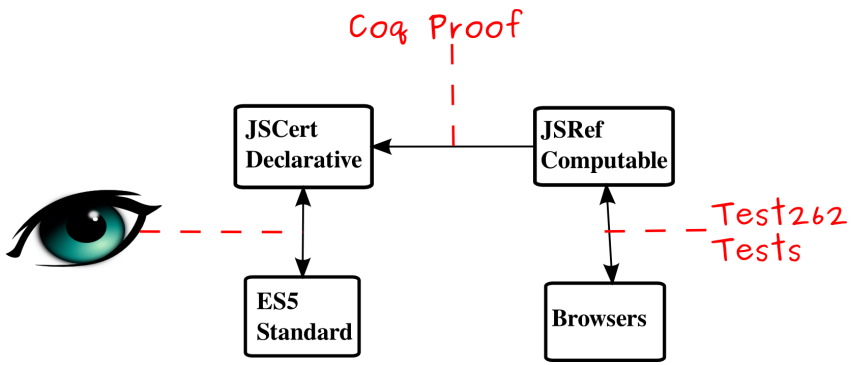
Imperial College London

Formal Methods Meets JavaScript Workshop
INRIA, Paris
23 March 2015

JSCert project: POPL'14

- JSCert: mechanised operational semantics for core ES5.
- JSRef : JS interpreter, proven correct against JSCert semantics.
- A basis for other analysis: e.g., sound type systems, program verification, flow analysis, abstract interpretation, ...

Establishing Trust



- "Eyeball" closeness of JSCert to ES5.
- Separate development of JSCert and JSRef.
- JSRef correct with respect to JSCert.
- Appropriate Test262 tests passed.

JSRef Testing Results: Paper

"JSRef successfully executes all the *tests we expect to pass* given our coverage of ES5"

	Chs 8-14			All test262		
	Pass	Fail	Abort	Pass	Fail	Abort
Paper results	1796	404	582	2919	3229	5598

Paper analysis of failed and aborted tests:

- For-in not implemented.
- Chapter 15 library functionality not implemented.
- Failures due to a non-conforming parser.

"We cannot yet guarantee that JSCert is bug free"

JSRef Testing Results: POPL'14

	Chs 8-14			All test262		
	Pass	Fail	Abort	Pass	Fail	Abort
Paper results	1796	404	582	2919	3229	5598
POPL results	1851	392	539	3090	3249	5406

Additional POPL changes:

- For loops implemented (28 tests).
- Argument object tests run (11 tests).
- hasOwnProperty method tests run (14 tests).
- Strict mode delete not throwing exception for unresolvable references (50 tests).

Analysis of Failed/Aborted Tests

Ch. 8–14 failed tests categorised to rule out expected failures. Two categories required attention:

- parser failures resulting in interpreter abort misattributed to external parser;
- some test failures were misattributed as being due to unimplemented features.

These categories previously over-approximated the set of permissable failures.

Chapter 15: the Array Library

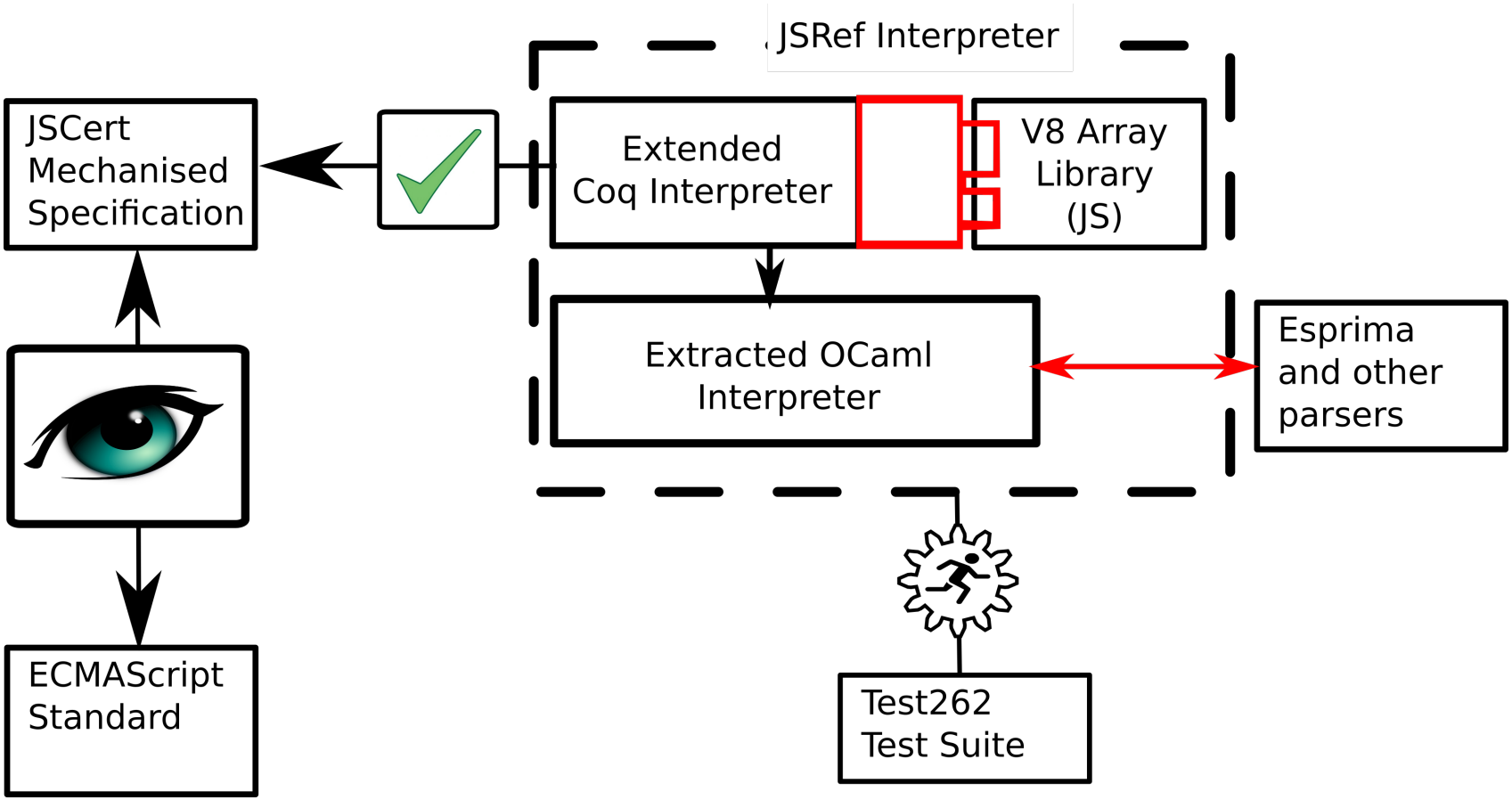
Aim: pass more tests by accounting for the Array Library.

Strategy: implement core Array functionality and use an existing self-hosted Array implementation.

Solution: use v8, as has clear separation of core functionality from self-hosted code.

Took opportunity to switch parser interface from Closure AST to SpiderMonkey standard AST (Esprima)

JSCert Project: the Array Library



JSRef Testing Results: Array Library

	Chs 8-14			All		
	Pass	Fail	Abort	Pass	Fail	Abort
Paper results	1796	404	582	2919	3229	5598
POPL results	1851	392	539	3090	3249	5406
Current	2435	131	216	4549	3366	3806
Current (+v8 Array)	2438	128	216	5481	2418	3822

This development process exposed a number of newly exposed bugs:

Bug	Category	Tests failed
For loop AST incorrect	Glue code	31
Eval not throwing Syntax Exception	Glue code	48
Floating point integer division incorrect	Coq float extraction	?
[[DefineOwnProperty]] helper logic	Shared Coq definition	?
Function syntax parsing	Parser	?
Eval in strict mode context	Weak specification	?

Future Work

- Port other self-hosted libraries: e.g., String, Boolean, Number,... (Cesar, the new Conrad, will do this over the summer.)
- Specify and prove self-hosted libraries using program logic. (Possibly using Daiva's work.)