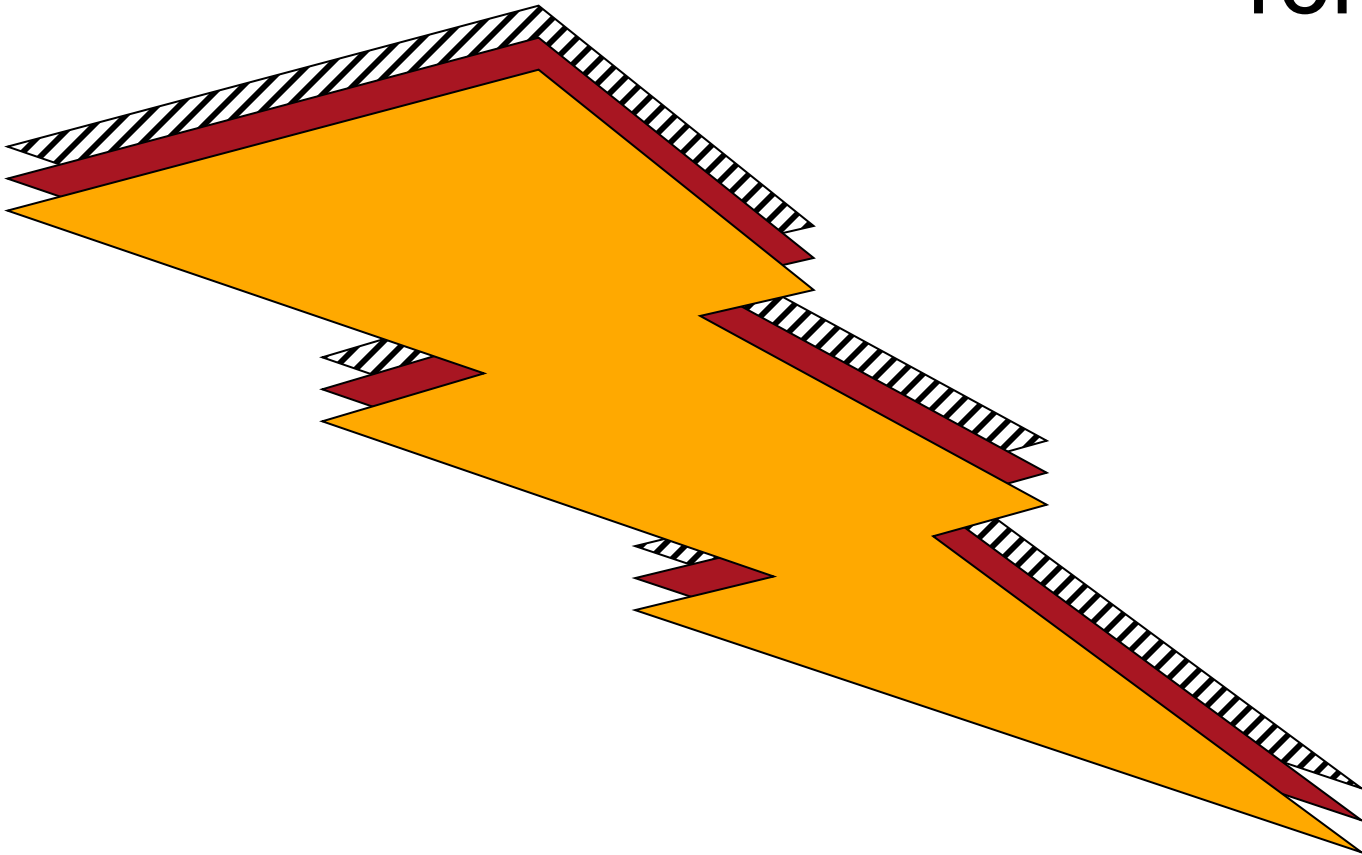


Verifying JavaScript and Creating Foundations for the Web



Shriram Krishnamurthi



JavaScript

("You got to dance with them what brung you")

Perspective on Semantics

Looking Ahead

Types to Verify JavaScript Programs

LAMBDA-CALCULUS MODELS OF PROGRAMMING LANGUAGES

James H. Morris

A system of types and type declarations is developed for the λ -calculus and its semantic assumptions are identified. The system is shown to be adequate in the sense that it permits a preprocessor to check formulae prior to evaluation to prevent type errors. It is shown that any formula

It is clear that the kind of undefinedness associated with nonterminating computations cannot be prevented if the language in question is a universal one. Our only aim is to provide for the undefinedness that arises from so-called don't-care conditions in language specifications.

Inferring Types in Smalltalk

Norihisa Suzuki
Xerox Palo Alto Research Centers
3333 Coyote Hill Rd., Palo Alto, CA 94304

Morris and Reynolds [9, 11] independently considered the same problem at about the same time. In typeless languages like lambda calculus (Morris) or Lisp (Reynolds), it is possible to encounter run-time errors such as applying lists to arguments. So the question that they posed is: Can one infer types of functions in these typeless languages, to catch more errors at compile time?

A Type Declaration and Inference System for Smalltalk

Alan H. Borning
Computer Science Dept., University of Washington

Daniel H. H. Ingalls
Xerox Palo Alto Research Center

machine-checkable documentation. While Smalltalk is a "type-safe" language in the sense that encountering an object of an inappropriate class will only result in a run-time error of the form "message not understood", it is nevertheless advantageous for the programmer to be informed of such a problem when the code in question is being compiled, rather than later when it is being used.

Retrofitted Type System Design Principle

Statically prevent (most)
existing run-time errors

`"a string" - "another string"`

`→ NaN`

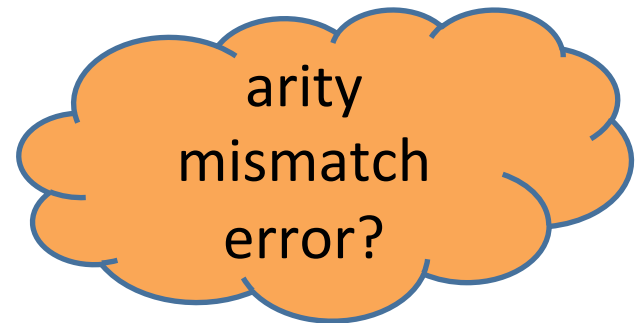
- Arithmetic doesn't signal errors
- No arity mismatch errors
- Reading non-existent field `→ undefined`
- Writing non-existent field `→` creates field
- Unbound identifiers `→` same story
- Breaching array bounds `→ undefined`
- Some object weirdness, too


```
var slice = function (arr, start, stop) {  
  
  var result = [];  
  for (var i = 0; i <= stop - start; i++) {  
    result[i] = arr[start + i];  
  }  
  return result;  
}
```

`slice([5, 7, 11, 13], 0, 2)`

→ `[5, 7, 11]`

`slice([5, 7, 11, 13], 2)`



stop: Num \cup Undef

```
var slice = function (arr, start, stop) {
```

```
  if (typeof stop === "undefined") {  
    stop = arr.length - 1;  
  }
```

stop: Undef

stop: Num

```
  var result = [];  
  for (var i = 0; i <= stop - start; i++) {  
    result[i] = arr[start + i];  
  }  
  return result;
```

stop: Num

```
}
```

```
  slice([5, 7, 11, 13], 0, 2)
```

```
→ [5, 7, 11]
```

```
  slice([5, 7, 11, 13], 2)
```

```
→ [11, 13]
```

Moral

“Scripting language” programmers
use state and
non-trivial control flow
to refine types

Typing Local Control and State using Flow Analysis

Arjun Guha, Claudiu Saftoiu, and Shriram Krishnamurthi

Brown University

Refactoring

Annotation

Program	LOC	Bad	Bozo	Auto	Man
analog_clock	112	0	6	13	0
animation	70	0	0	4	1
catchit	165	7	9	6	3
countdown	129	2	12	4	0
hashapass	257	1	7	13	7
light	151	8	19	3	7
metronome	106	1	4	10	2
morse	275	8	5	12	0
resistor	591	18	2	32	0
rsi	328	0	27	22	0
text2wav	488	3	6	38	3
topten	443	67	0	18	0
watchtimer	947	18	7	15	2
TOTAL	4062	133	104	190	25

A Completely Different Application

GREAT ESCAPES TO EUROPE.

Continental Airlines

The New York Times

Tuesday, June 1, 2010 Last Update: 8:40 PM ET

Fares from \$407 each way

Taxes and fees apply

Continental Airlines

Switch to the Global Edition for an international perspective on news, business, sports and more.

Search



Subscribe to Home Delivery | Personalize Your Weather

Switch to Global Edition

JOBS REAL ESTATE AUTOS ALL CLASSIFIEDS

WORLD
U.S.
POLITICS
N.Y./REGION
BUSINESS
TECHNOLOGY
SPORTS
SCIENCE
HEALTH
OPINION
ARTS
Books
Movies
Music
Television
Theater
STYLE
Dining & Wine
Fashion & Style
Home & Garden
Weddings/
Celebrations

After Israel Raids Flotilla, U.S. Is Torn Between Allies

By MARK LANDLER 15 minutes ago

The rift between Israel and Turkey makes it difficult for the White House to make progress on Iran's nuclear program and peace talks between Israel and the Palestinians.

U.N. Council Condemns 'Acts' in Raid 12:18 PM ET

- MORE ON THE RAID
- Israel Faces Pressure on Gaza After Raid
 - Room for Debate: Rethinking a Blockade
 - The Lede: Activists Pledge to Send More Ships to Gaza

U.S. Opens



Juan Arredondo for The New York Times

Immigrant Runs for Mayor, Back in Mexico

By KIRK SEMPLE 15 minutes ago

Juan Navarro has homes in Queens and New Jersey, but his electoral goal is the mayor's office in Serdan, Mexico. Above, Mr. Navarro at his restaurant in Manhattan.

Half a Dozen States Delay Tax Refunds

By MICHAEL COOPER 13 minutes ago

Some states are cash poor, and others also lack the ability

OPINION »

Op-Ed: Israeli Force, Adrift on the Sea

After the botched raid on the Gaza flotilla, Israel must accept that power can never defeat an idea, writes Amos Oz. Instead, what is needed is a better idea.

- Brooks: The Oil Plume
- Comments (560)
- Herbert: Epic Foolishness
- Editorial: Habeas Corpus
- O'Hanlon et al.: War Data
- Revkin: Gulf Crimes?
- Room for Debate: Raise the Retirement Age?

TRAVEL »

Caravaggio in Rome

About a third of the artist's works are housed in Rome.



MARKETS »

At 8:21 PM ET

JAPAN	Nikkei	HangSeng	CHINA	Shanghai
9,632.83	-79.00	19,496.95	2,568.28	-23.86
-0.81%		-1.36%		-0.92%

Data delayed at least 15 minutes

GET QUOTES

My Portfolios | Stock, ETFs, Funds | Go

Will it be worth a

June 26, 2012

HUFFPOST BUSINESS

THE INTERNET NEWSPAPER: NEWS BLOGS VIDEO COMMUNITY

Edition: U.S.

Search The Huffington Post

Like 33k Follow +1

FRONT PAGE POLITICS ENTERTAINMENT WORLD TECH MEDIA GREEN SPORTS SCIENCE CULTURE ALL SECTIONS

Business > Small Business Money The Watchdog Occupy Wall Street

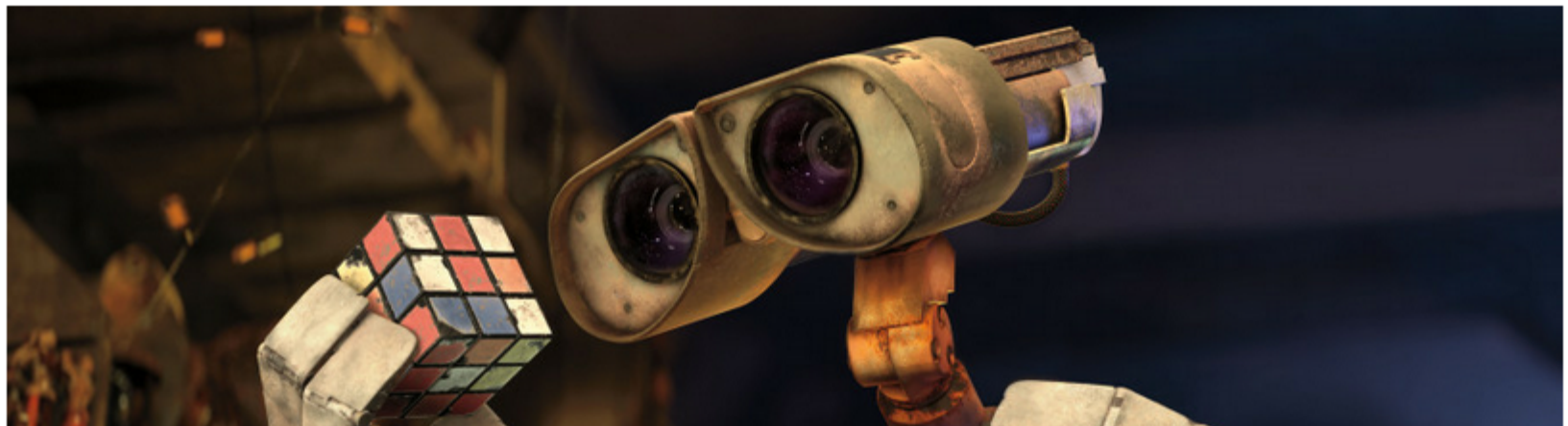
TechCrunch Autoblog

FROM AP: Congress passes bill increasing drug inspections... 1 hour 24 minutes ago

Enter email address Get Alerts

WALL-E STREET

Humans Band Together To Fight Back The Rise Of The Machines



```
// Redirect page  
window.location = "citibank.com.evil.com"
```

```
// Change all links  
links = document.getElementsByTagName("a");  
for (var i = 0; i < links.length; i++) {  
    links[i].href = "track.com/fwd?" +  
links[i].href; }  

```

```
// Read cookies  
document.cookie
```

```
// Read passwords  
document.querySelector('input[type=password]')
```

```
// Embed Flash, exploit, profit  
document.write(`  
    <object type="application/x-shockwave-flash"  
        data="evil.swf" />`);
```




Facebook
JavaScript (FBJS)



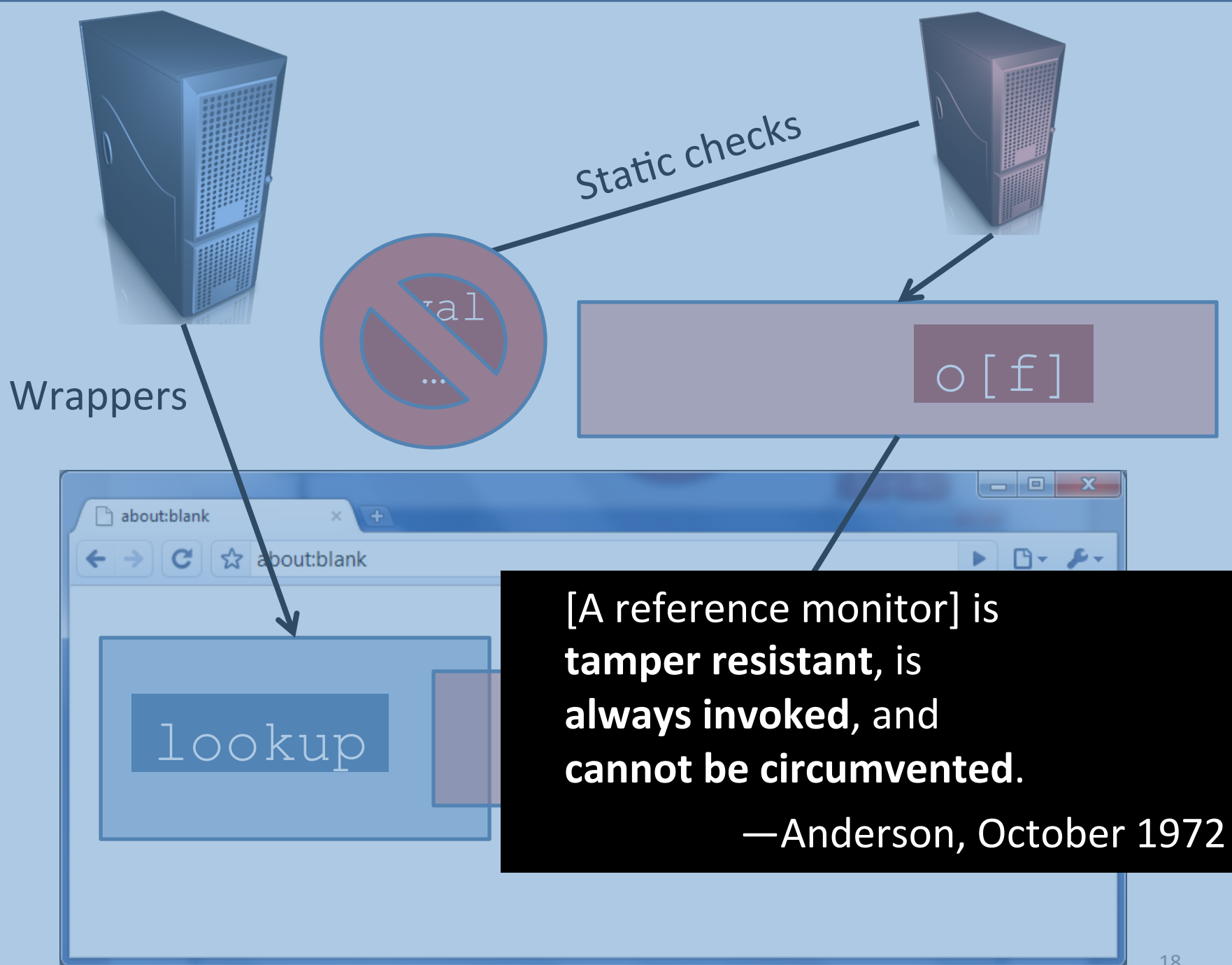
Microsoft
Web Sandbox



Google
Caja



Yahoo!
ADsafe



I need your help in testing its robustness. Are the rules sufficient to prevent all direct access to the DOM and the global object? Are there any small leaks that I am unaware of? Is the approach I'm taking inherently unsound? What additional restrictions are required to prevent unintended collusion?

So this is the test:

Write a program in the form

```
(function () {  
  ...  
})();
```



where the ... is replaced by code that calls the alert function when run on any browser. If the program produces no errors when linted with the ADsafe option, then I will buy you a plate of shrimp.

caplet list, 2007-09-30

Type-check
the body of
`adsafe.js`

JSlint rejects

JSlint passes

```
adsafe.js
ADSAFE = {
  get: function(),
  set: function(),
  ...};
```

```
ad.js
ADSAFE.get
ADSAFE.set (o
...
```

Encode all of
JSlint as
a type



The Need for Flexibility

ADsafety
Type-Based Verification of JavaScript Sandboxing

Joe Gibbs Politz Spiridon Aristides Eliopoulos Arjun Guha Shriram Krishnamurthi

Brown University

**Verifying Web Browser Extensions' Compliance
with Private-Browsing Mode**

Benjamin S. Lerner, Liam Elberty, Neal Poole, and Shriram Krishnamurthi

Brown University

**Combining Form and Function:
Static Types for JQuery Programs***

Benjamin S. Lerner, Liam Elberty, Jincheng Li, and Shriram Krishnamurthi

Brown University

TeJaS: Type **Systems** for JavaScript

Benjamin S. Lerner

Brown University
blerner@cs.brown.edu

Joe Gibbs Politz

Brown University
joe@cs.brown.edu

Arjun Guha

Cornell University
arjun@cs.cornell.edu

Shriram Krishnamurthi

Brown University
sk@cs.brown.edu

5.2 Example: Implementing TypeScript's Covariant Function Calls

As a proof of concept, we have implemented an extension to provide TypeScript's semantics for functions [18]. This extension overrides the `TArrow` type of our base system, and replaces it with one that has the new semantics. The types-definition module is gratifyingly similar to the `Base` one: the only change necessary is adding a single type constructor

```
1   type typ =  
2     | TBase of BASE.typ  
3     | TArrow of typ list * typ option * typ
```

In fact, the entire extension is only 1860LOC: other than minor naming-convention differences, the 260-line difference between the two is precisely that which defines how TypeScript's arrow types behave.

Semantic Foundations

The Essence of JavaScript

Arjun Guha, Claudiu Saftoiu, and Shriram Krishnamurthi

Brown University

$c = \text{num} \mid \text{str} \mid \text{bool} \mid \text{undefined} \mid \text{null}$
 $v = c \mid \text{func}(x \dots) \{ \text{return } e \} \mid \{ \text{str} : v \dots \}$
 $e = x \mid v \mid \text{let } (x = e) e \mid e(e \dots) \mid e[e] \mid e[e] = e \mid \text{de}$
 $E = \bullet \mid \text{let } (x = E) e \mid E(e \dots) \mid v(v \dots E, e \dots)$
 $\mid \{ \text{str} : v \dots \text{str} : E, \text{str} : e \dots \} \mid E[e] \mid v[E] \mid E[e]$
 $\mid v[v] = E \mid \text{delete } E[e] \mid \text{delete } v[E]$

$\text{let } (x = v) e \hookrightarrow e[x/v] \dots$

$(\text{func}(x_1 \dots x_n) \{ \text{return } e \})(v_1 \dots v_n) \hookrightarrow e[x_1/v_1 \dots]$

$\{ \dots \text{str} : v \dots \}[\text{str}] \hookrightarrow v$

$\frac{\text{str}_x \notin (\text{str}_1 \dots \text{str}_n)}{\{ \text{str}_1 : v_1 \dots \text{str}_n : v_n \} [\text{str}_x] \hookrightarrow \text{undefined}}$ (E-C)

$\{ \text{str}_1 : v_1 \dots \text{str}_i : v_i \dots \text{str}_n : v_n \} [\text{str}_i]$
 $\hookrightarrow \{ \text{str}_1 : v_1 \dots \text{str}_i : v \dots \text{str}_n : v_n \}$

$\frac{\text{str}_x \notin (\text{str}_1 \dots)}{\{ \text{str}_1 : v_1 \dots \} [\text{str}_x] = v_x \hookrightarrow \{ \text{str}_x : v_x, \text{str}_1 : v_1 \dots \}}$

$\text{delete } \{ \text{str}_1 : v_1 \dots \text{str}_i : v_x \dots \text{str}_x : v_n \} [$
 $\hookrightarrow \{ \text{str}_1 : v_1 \dots \text{str}_i : v \dots \text{str}_n : v_n \}$

$\frac{\text{str}_x \notin (\text{str}_1 \dots)}{\text{delete } \{ \text{str}_1 : v_1 \dots \} [\text{str}_x] \hookrightarrow \{ \text{str}_1 : v_1 \dots \}}$ (E-DE)

Fig. 1. Functions and Objects

λ_{JS} (sort of)
on one slide

$l = \dots$ Locations
 $v = \dots \mid l$ Values
 $\sigma = (l, v) \dots$ Stores
 $e = \dots \mid e = e \mid \text{ref } e \mid \text{deref } e$ Expressions
 $E = \dots \mid E = e \mid v = E \mid \text{ref } E \mid \text{deref } E$ Evaluation Contexts

$\frac{e_1 \hookrightarrow e_2}{\sigma E \langle e_1 \rangle \rightarrow \sigma E \langle e_2 \rangle}$

$\frac{l \notin \text{dom}(\sigma) \quad \sigma' = \sigma, (l, v)}{\sigma E \langle \text{ref } v \rangle \rightarrow \sigma' E \langle l \rangle}$ (E-REF)

$\sigma E \langle \text{deref } l \rangle \rightarrow \sigma E \langle \sigma(l) \rangle$ (E-DEREF)

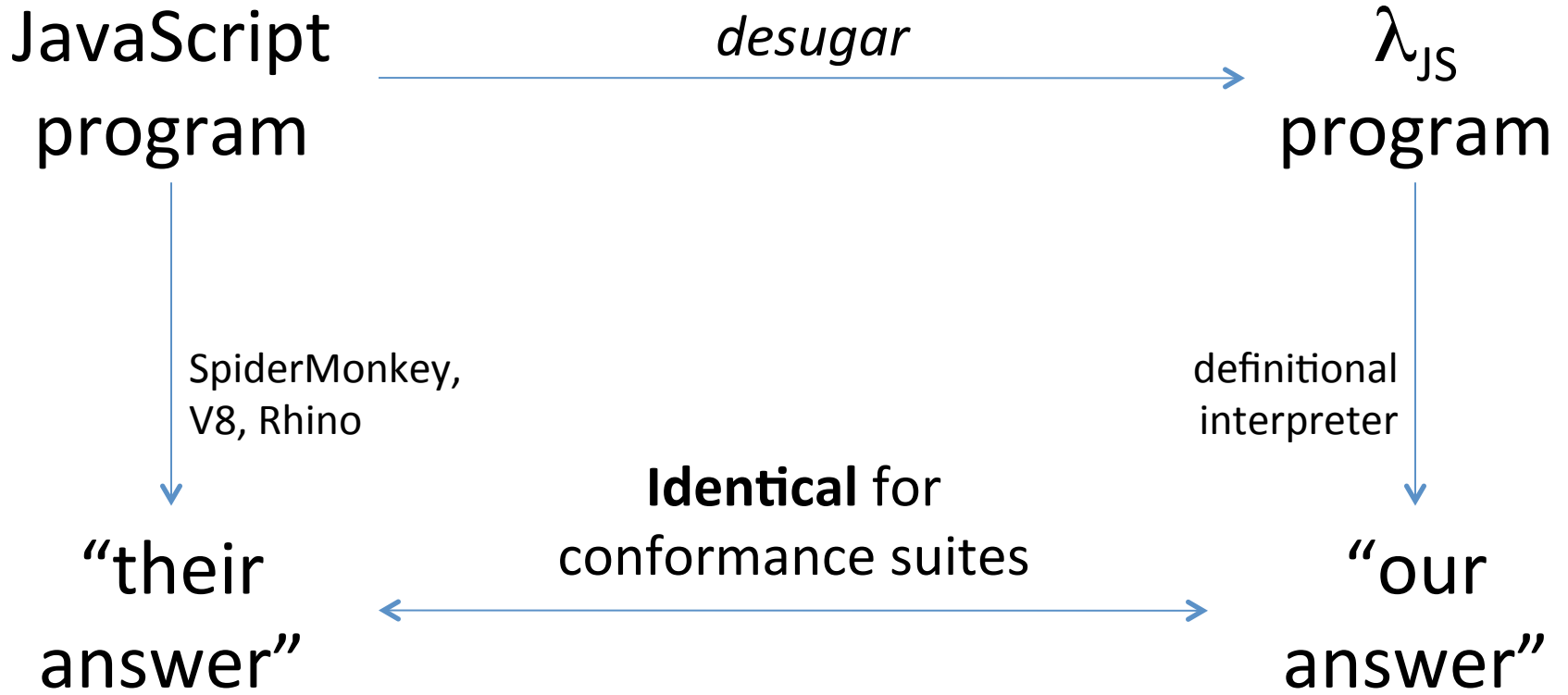
$\sigma E \langle l = v \rangle \rightarrow \sigma[l/v] E \langle l \rangle$ (E-SETREF)

$\frac{\text{str}_x \notin (\text{str}_1 \dots \text{str}_n) \quad \text{"_proto_"} \notin (\text{str}_1 \dots \text{str}_n)}{\{ \text{str}_1 : v_1, \dots, \text{str}_n : v_n \} [\text{str}_x] \hookrightarrow \text{undefined}}$ (E-GETFIELD-NOTFOUND)

$\frac{\text{str}_x \notin (\text{str}_1 \dots \text{str}_n)}{\{ \text{str}_1 : v_1 \dots \text{"_proto_"} : \text{null} \dots \text{str}_n : v_n \} [\text{str}_x] \hookrightarrow \text{undefined}}$
 (E-GETFIELD-PROTO-NULL)

$\frac{\text{str}_x \notin (\text{str}_1 \dots \text{str}_n) \quad p = \text{ref } l}{\{ \text{str}_1 : v_1 \dots \text{"_proto_"} : p \dots \text{str}_n : v_n \} [\text{str}_x] \hookrightarrow (\text{deref } p) [\text{str}_x]}$
 (E-GETFIELD-PROTO)

Fig. 4. Prototype-Based Objects



**A Tested Semantics for
Getters, Setters, and Eval in JavaScript**

Joe Gibbs Politz Matthew J. Carroll Benjamin S. Lerner Justin Pombrio Shriram Krishnamurthi
Brown University
www.jswebtools.org



- *JavaScript Verification and Full Abstraction*, MSR
- *System !D*, UCSD
- *Aspects for JavaScript*, U Chile
- *Formal Specification of JavaScript Modules*, KAIST
- *JavaScript Abstract Machine*, Utah and Northeastern
- *Deriving Refocusing Functions*, Aarhus
- *Information Flow Analysis*, Stevens Tech
- *OCFA*, Fujitsu Labs (patent pending)

PERSPECTIVE ON SEMANTICS

Modeling and Reasoning about DOM Events

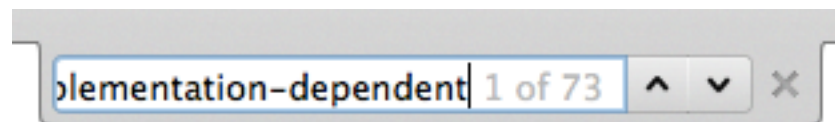
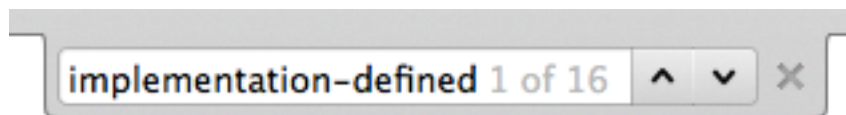
Benjamin S. Lerner Matthew J. Carroll Dan P. Kimmel
Hannah Quay-de la Vallee Shriram Krishnamurthi
Brown University



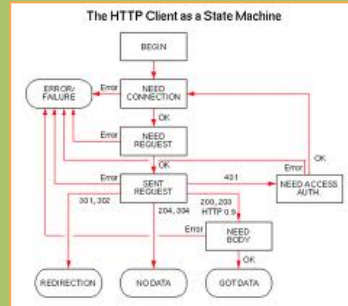
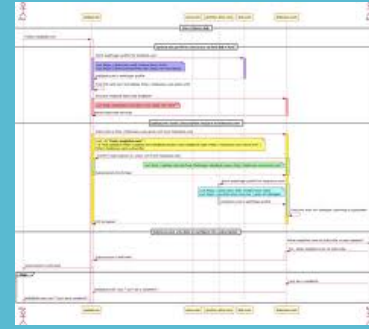
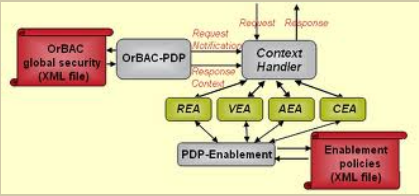
What About the Spec?

1. The spec is embodied in the implementations.
2. The spec is incomplete: e.g., SES depends on **window.console**
3. The spec depends on implementations!

*If [...], the behavior of **sort** is *implementation-defined*.*



4. Attackers attack implementations, not specs.



Semantics as Mathematics
Semantics as Natural Science

(Reality might be stranger than we expect)



LOOKING AHEAD

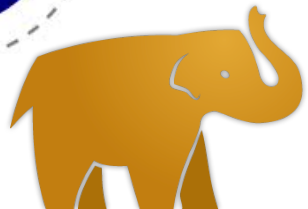
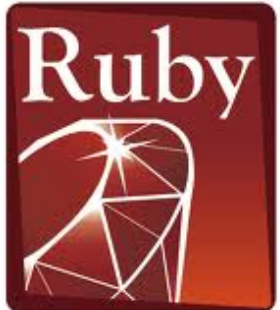
JavaScript Types and Semantics

```
graph TD; A[JavaScript Types and Semantics] --> B[Continuing to Embrace Current Systems]; A --> C[Foundations for Next-Generation Systems];
```

Continuing
to Embrace
Current Systems

Foundations for
Next-Generation
Systems

How Many Languages?




compiler - Documentation o x

stackoverflow.com/questions/789093/documentation-on-creating-a-programming-language

My Calendar My Papers Gandi ASK Conferences Journals Brown Travel Courses Other Bookmarks

StackExchange v log in careers 2.0 chat meta about faq search

 **Questions** Tags Users Badges Unanswered **Ask Question**


Documentation on creating a programming language

C++ - create my own progra x

stackoverflow.com/questions/3662410/create-my-own-programming-language

My Calendar My Papers Gandi ASK Conferences Journals Brown Travel Courses Other Bookmarks

StackExchange v log in careers 2.0 chat meta about faq search

 **Questions** Tags Users Badges Unanswered **Ask Question**

create my own programming language [closed]

How to go about making you x

stackoverflow.com/questions/3810119/how-to-go-about-making-your-own-programming-language

My Calendar My Papers Gandi ASK Conferences Journals Brown Travel Courses Other Bookmarks

StackExchange v log in careers 2.0 chat meta about faq search

 **Questions** Tags Users Badges Unanswered **Ask Question**

How to go about making your own programming language? [closed]

Research Challenge

Apply machine learning
to learn the semantics
of programming languages
(and **libraries** and **frameworks** and ...)

JavaScript Types and Semantics

```
graph TD; A[JavaScript Types and Semantics] --> B[Continuing to Embrace Current Systems]; A --> C[Foundations for Next-Generation Systems];
```

Continuing
to Embrace
Current Systems

Foundations for
Next-Generation
Systems



Synthesis

Alchemy: Transmuting Base Alloy Specifications into Implementations

Shriram Krishnamurthi
Brown University

Kathi Fisler
WPI

Daniel J. Dougherty
WPI

Daniel Yoo
WPI

Towards an Operational Semantics for Alloy

Theophilos Giannakopoulos,¹ Daniel J. Dougherty,¹
Kathi Fisler,¹ Shriram Krishnamurthi²

¹ Department of Computer Science, WPI

² Computer Science Department, Brown University

Toward a More Complete Alloy ^{*,**}

Timothy Nelson¹, Daniel J. Dougherty¹, Kathi Fisler¹, and Shriram Krishnamurthi²

¹ Worcester Polytechnic Institute

² Brown University

Aluminum: Principled Scenario Exploration through Minimality

Tim Nelson¹, Salman Saghaei¹, Daniel J. Dougherty¹, Kathi Fisler¹, Shriram Krishnamurthi²

¹Department of Computer Science
WPI

Lessons for Language Design

A Simple Challenge

```
def f(x):  
    class C(object):  
        x = "C's x"  
        def meth(self):  
            return x + ', ' + C.x  
    return C  
  
f('input x')().meth()
```



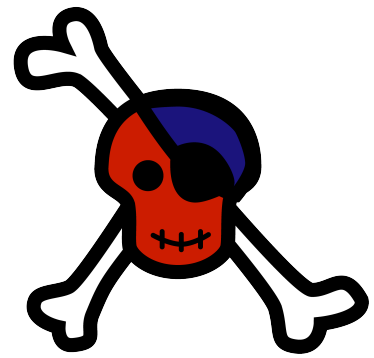
High-Level Problem

Scripting languages have evolved to have:

- over-blurring of objects vs. dictionaries
- awful scoping rules
- hostility to static types

All three traits are **antithetical to verification**

Pyret



Preserves the **essence of scripting**

IDE-Friendly: Clean scope and types/contracts

Two novel ideas for programming languages:

dependent mixins

relational object types

www.jswebtools.org

JavaScript Types and Semantics

Continuing
to Embrace
Current Systems

Foundations for
Next-Generation
Systems

