An Empirical Study on the Rewritability of the with Statement in JavaScript

Motivation

- Provide empirical data for with statements used in real-world JavaScript applications
- Amount
- Usage patterns
- Check rewritability of with statements in real world
- Rewriting rules to remove with statements by replacing them with other statements

Introduction of the with statement

- Syntax and semantics in ECMAScript
- Provide empirical data for with statements

Syntax

```javascript
with(x) { ... }
with(x) { ... with y }
```

Semantics

1. Evaluate `x` as a JavaScript object
2. Add `y` to the front of the scope chain
3. Evaluate `y`
4. Remove `y` from the scope chain

The `with` statement introduces a new scope at runtime.

Examples

```javascript
with(document.body.children[0].style){
  textAlign="center";
}
```

Good Parts

- Provide a convenient way to develop dynamically changing web contents

Examples

```html
<html>
  <body>
    ... <div id="body"> ... </div> ... </body>
</html>
```

Bad Parts

- Incur performance overheads
- Make static analysis infallible

Examples

```javascript
var obj1={onem:1};
var obj2={twon:2};
function fun1(obj1){
  obj1.onem=0;
  with(obj1)
  ... obj.id ...
}
```

Rewritability

- Defining the rewrite function
- Program without statement

Rewriting Rules

```javascript
rewriting x:with(y){ z }
```

Rewriting the Assignment

```javascript
var $f=withObject(obj);
```

Rewritability Check

- We can rewrite all static with statements in all patterns except for 4 patterns.

Goal

- Can we answer the question at static time?

```javascript
with(obj) { ... id ... }
```

Main Idea

- `id` has the `id` field?
- `foo(x)`
- `this.publicF` instead of `this.F`?
- `new F({})` instead of `new F()`?
- `with(obj)`

Rewriting the Identifier

```javascript
with(obj) { ... id ... }
```

Identifiers are rewritten to ternary conditional expressions

```javascript
rewriting x:with(y){ z }
```

- `a` in `$f$ ? $f$.a : $f$` to `$a$` in `${f.a}[f.a] ? f.a : f.a`