

2013W1-lecture6

September 16, 2013

Contents

1	Question of the Day	1
1.1	Solution	1
2	Logistics	2
2.1	Programming Assignment #2	2
2.2	Conceptual Assignment #1 (quiz corrections) due Tue at 8PM	2
2.3	Reminder of Piazza (e.g., to answer “Where do I find <code>plai</code> -typed documentation?”)	3
3	Desugaring	3
4	What have we learned today?	4

1 Question of the Day

What happens in Racket when we add $2^{30} + 2^{30}$?

Let’s also try it in C:

```
printf("%d", 1073741824 + 1073741824);  
  
-2147483648.0
```

1.1 Solution

Oh, so you think whatever semantics Racket provides is perfect? Try this, smarty-pants:

```

; Multiply k by the fraction f until it reaches zero.
(define (reduce-to-zero k f)
  (if (= k 0.0)
      true
      (reduce-to-zero (* f k) f)))

; Which of these terminates?
(reduce-to-zero 0 0)
(reduce-to-zero 1 0.25)
(reduce-to-zero 1 0.75)
(reduce-to-zero 1 1)

```

Now, imagine that we created our own Racket implementation of the IEEE floating-point standard (perhaps with bugs) and wrote the same function. Is it easy to determine statically whether this terminates?

In fact, there *are* static analysis systems that (maybe) can answer this question in this case because people have worked *hard* to find analyses that are effective for these sorts of questions.

But you will:

1. theoretically always be able to come up with a question that cannot be answered statically for some programs and
2. practically always be able to come up with a question that cannot be answered statically by the (sufficiently efficient, perhaps) available static analyses.

2 Logistics

2.1 Programming Assignment #2

Out today!

Teams of 2-3. Find a team! (We *strongly prefer* pairs.)

Due one week from Friday.

Start early b/c we may release another programming assignment next Monday due the following Friday!

2.2 Conceptual Assignment #1 (quiz corrections) due Tue at 8PM

See the quiz document itself (posted in the assignments area of the website). The first one is handin target `ca1`.

2.3 Reminder of Piazza (e.g., to answer “Where do I find plai-typed documentation?”)

<https://piazza.com/ubc.ca/winterterm12013/cpsc311>

WARNING: I will **close** Piazza registration by Friday!

3 Desugaring

Let’s add functionality to our non-deterministic calculator using desugaring!

Desugaring is the process of converting a surface language to an underlying core via a static transformation. In other words, we’re not running the program yet, but we are “simplifying” it.

Be careful of that word “simplifying”, however. In some cases, the underlying core may seem more complex than the surface language, but it will generally present at least one of the advantages below.

So, why would we want to desugar?

BRAINSTORM!

- allows us to build different surface languages atop a single core
- may make either the program or the interpreter faster
- better grasp of semantics in “leaner” core language
- separation of concerns: don’t worry simultaneously about user interface needs and language design needs
- customize the language to the domain at hand
- simpler maintenance for core language
- keep the same surface syntax but improve the underlying language (backwards compatibility)

SOLUTION

At least:

- + present the programmer with a "friendlier" interface,
- + build a different looking language atop an existing core,
- + reduce a complex language to a simpler one to make our language implementation job easier,
- + reduce a complex language to a simple core to make it easier to

reason about the core (e.g., to prove security, correctness, or resource use properties),
+ add desired features to the language without complicating its implementation.

See file:2013W1-desugaring-nondet-interp.rkt.

4 What have we learned today?

From QotD:

- More on the distinction between “static” and “dynamic”.
- Desugaring
 - Explain the relationship of a desugarer to the parser and interpreter.
 - Justify the use of desugaring on various grounds, including its impact on the programmer’s interface(s), the language implementation’s complexity, the tractability (doable-ness) of useful reasoning over the language, and the software engineering advantages of isolating components of the interpreter system.
 - Extend a surface language over an existing core language using a desugaring.