Semantics of Programming Languages

Renato Neves

Universidade do Minho

Table of Contents

The search for meaning

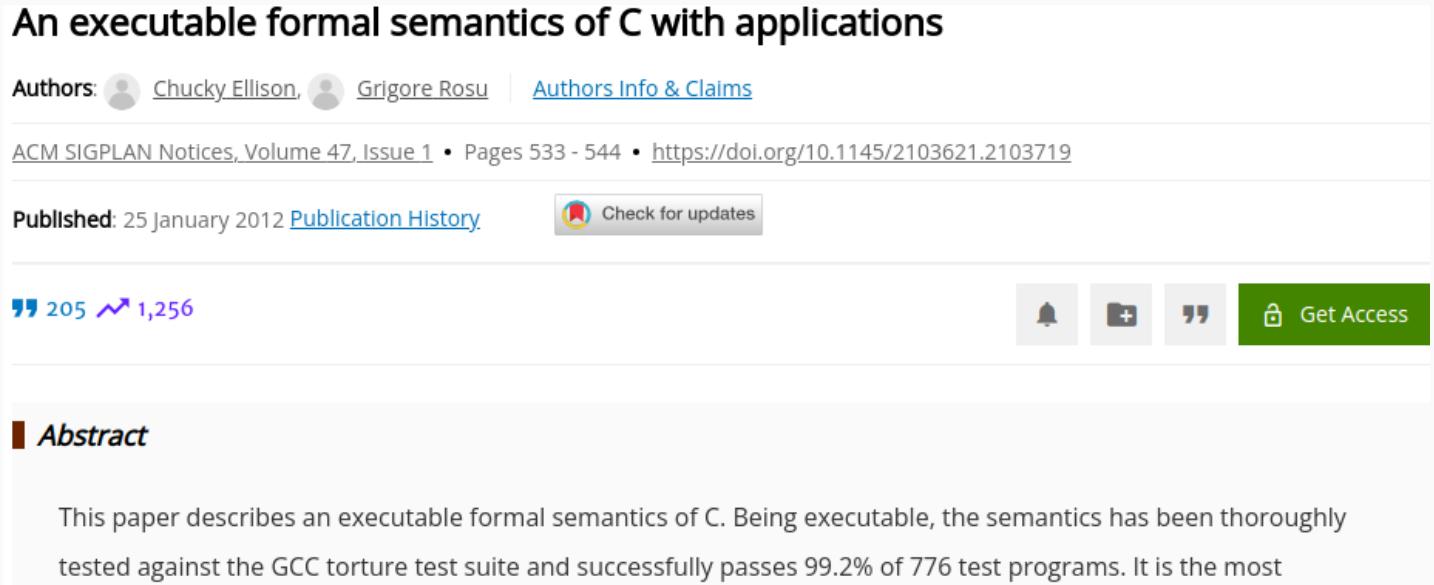
Examples

- $f()\{\texttt{prt}\ a;\ \texttt{ret}\ 1\};\ g()\{\texttt{prt}\ b;\ \texttt{ret}\ 0\};\ x := f() \vee g()$
- $x := 2\ ;(x := x + 1 \parallel x := 0)$
- $(a := 1\,;\texttt{prt}\ b) \parallel (b := 1\,;\texttt{prt}\ a)$

The search for meaning

Examples

- $f()\{\texttt{prt}\ a;\ \texttt{ret}\ 1\};\ g()\{\texttt{prt}\ b;\ \texttt{ret}\ 0\};\ x := f() \vee g()$
- $x := 2\ ;(x := x + 1 \parallel x := 0)$
- $(a := 1\,;\texttt{prt}\ b) \parallel (b := 1\,;\texttt{prt}\ a)$



An executable formal semantics of C with applications

Authors: Chucky Ellison, Grigore Rosu | Authors Info & Claims

ACM SIGPLAN Notices, Volume 47, Issue 1 • Pages 533 - 544 • https://doi.org/10.1145/2103621.2103719

Published: 25 January 2012 Publication History | Check for updates

205 1,256 | Get Access

Abstract

This paper describes an executable formal semantics of C. Being executable, the semantics has been thoroughly tested against the GCC torture test suite and successfully passes 99.2% of 776 test programs. It is the most

The search for meaning

Examples

- $\mathtt{p}; (\mathtt{q}; \mathtt{r}) \stackrel{?}{=} (\mathtt{p}; \mathtt{q}); \mathtt{r}$
- $\mathtt{p} \parallel \mathtt{q} \stackrel{?}{=} \mathtt{q} \parallel \mathtt{p}$
- $\left(\mathtt{p} +_{\frac{1}{2}} \mathtt{q}\right); \mathtt{r} \stackrel{?}{=} \mathtt{p}; \mathtt{r} +_{\frac{1}{2}} \mathtt{q}; \mathtt{r}$
- $\mathtt{entangle}(\mathtt{x}, \mathtt{y}) \stackrel{?}{=}$ spooky action

The search for meaning

Examples

- $\mathtt{p}; (\mathtt{q}; \mathtt{r}) \stackrel{?}{=} (\mathtt{p}; \mathtt{q}); \mathtt{r}$
- $\mathtt{p} \parallel \mathtt{q} \stackrel{?}{=} \mathtt{q} \parallel \mathtt{p}$
- $\left(\mathtt{p} +_{\frac{1}{2}} \mathtt{q}\right); \mathtt{r} \stackrel{?}{=} \mathtt{p}; \mathtt{r} +_{\frac{1}{2}} \mathtt{q}; \mathtt{r}$
- $\mathtt{entangle}(\mathtt{x}, \mathtt{y}) \stackrel{?}{=}$ spooky action

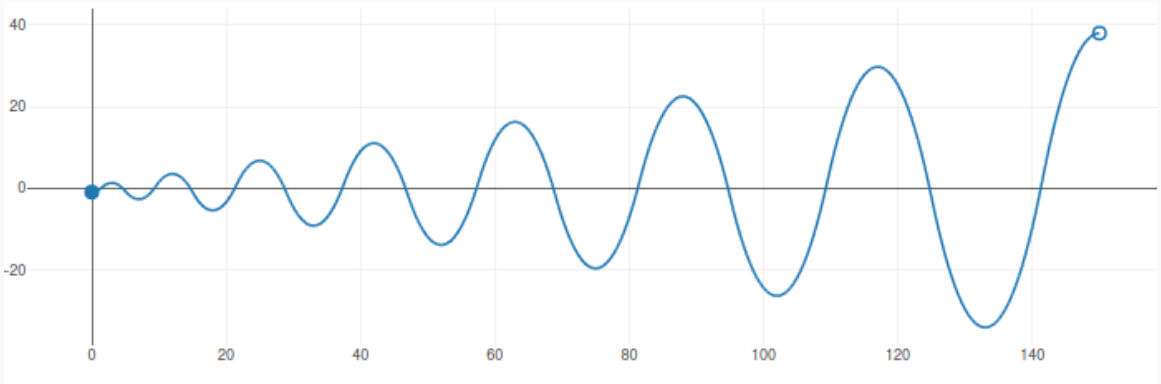
Will my program behave correctly ?

A particle and its orbital trajectory – what can go wrong?

```
x := -1; v := 0; a := 1;
while true do {
      if x <= 0 then a := 1 else a :=-1;
      x' = v, v' = a for 0.5;
}
```

A particle and its orbital trajectory – what can go wrong?

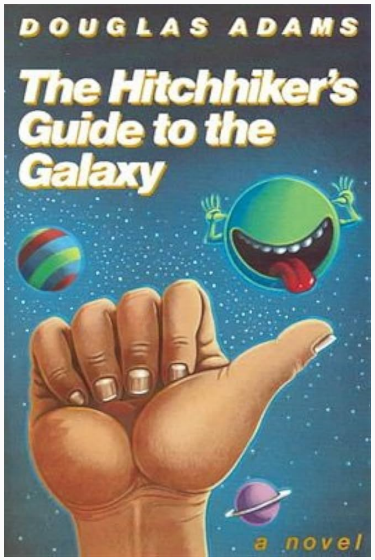
```
x := -1; v := 0; a := 1;
while true do {
      if x <= 0 then a := 1 else a :=-1;
      x' = v, v' = a for 0.5;
}
```



The search for meaning

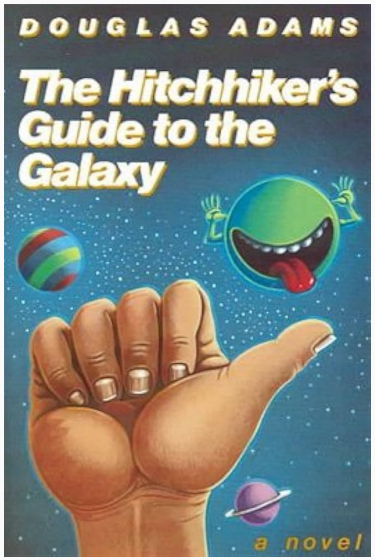
But. . . what is a programming language really ?

And what is computation ?



The search for meaning

But. . . what is a programming language really ?
And what is computation ?



To answer these questions we will turn programming into mathematics . . .

in the same way that (other?) natural phenomena are turned into mathematics (electromagnetism, quantum physics . . .)

Table of Contents

A sprinkle of linguistics

We will face two linguistic concepts that every programmer ought to know

- syntax - determines whether a sentence is valid or not
- semantics - the meaning of valid sentences

Example (syntax)

The sentence (program) $\mathtt{x} := \mathtt{p}\,;\mathtt{q}$ is forbidden by the syntactic rules of most programming languages

Example (semantics)

The sentence (program) $\mathtt{x} := 1$ has the meaning "writes 1 in the memory address corresponding to $\mathtt{x}$"

Semantics for every season

Operational semantics	How a program operates
Denotational semantics	What a program is
Axiomatic semantics	Which logical properties a program satisfies

Semantics for every season

Operational semantics	How a program operates
Denotational semantics	What a program is
Axiomatic semantics	Which logical properties a program satisfies

Some truly surprising properties arise from their interaction . . .

An apparently impossible program

Take a concurrent program `p`

It relies on an external scheduler to interleave its actions

There are infinitely many schedulers even uncountably many

Yet one can write down a program that checks whether `p` satisfies a given property for all schedulers ... in finite time !!

How deep will we go into the rabbit hole ?

Our learning path will intersect theory and practice, from the very basics to the state-of-the-art — we will face current limitations and see what challenges lie ahead

Table of Contents

Assessment

Two written tests (24 Mar and 26 May)

Materials and Contacts

Relevant class material and announcements posted on the website

`http://lmf.di.uminho.pt/SLP-2425/`

e-mail: nevrenato@di.uminho.pt

office hours: wednesday afternoon (please send an email the day before if you wish to meet)


Roy L. Crole, *Categories for types*, Cambridge mathematical textbooks, Cambridge University Press, 1993.


Hanne Riis Nielson and Flemming Nielson, *Semantics with applications: An appetizer*, Springer London, 2007.


Glynn Winskel, *The formal semantics of programming languages - an introduction*, Foundation of computing series, MIT Press, 1993.