ProCoS: How It All Began – as seen from Denmark

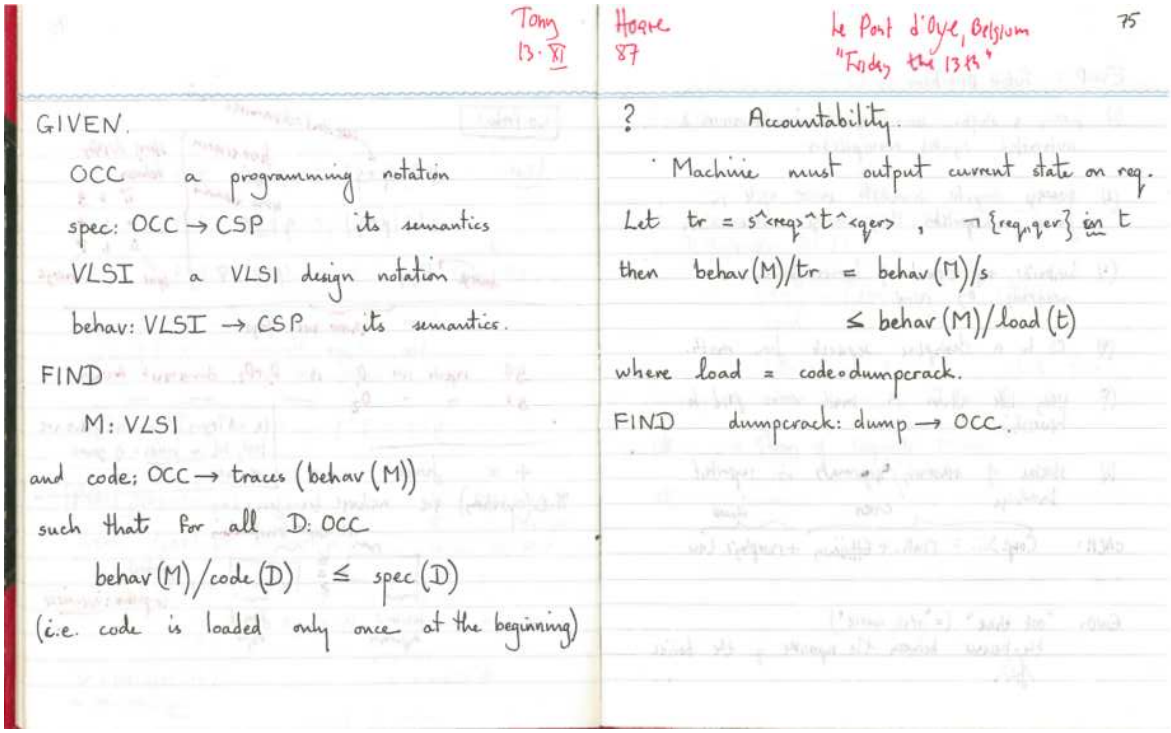
Dines Bjørner
Fredsvej 11, DK-2840 Holte, Denmark*

October 30, 2015

Abstract

I reminisce over an episode at the 9–13 November 1987 IFIP WG2.3 meeting at Château du Pont d'Oye in Belgium — and at what followed.

I had given a half hour presentation of how we, in Denmark, had developed a compiler for the full Ada programming language. My presentation had evolved around a single slide showing boxes and arrows between these, all properly labeled. Edsger W. Dijkstra had railed during my short presentation against my using diagrams — despite my claiming that boxes denoted certain kinds of algebras and arrows certain kind of morphisms between these. After my talk there was a coffee break. Tony Hoare took me aside. Asked permission to write in my note book. And this is what he wrote:


Tony 13.XI

GIVEN.
OCC a programming notation
spec: OCC → CSP its semantics
VLSI a VLSI design notation
behav: VLSI → CSP its semantics.
FIND
M: VLSI
and code: OCC → traces (behav (M))
such that for all D: OCC
behav (M)/code (D) ≤ spec (D)
(i.e. code is loaded only once at the beginning)

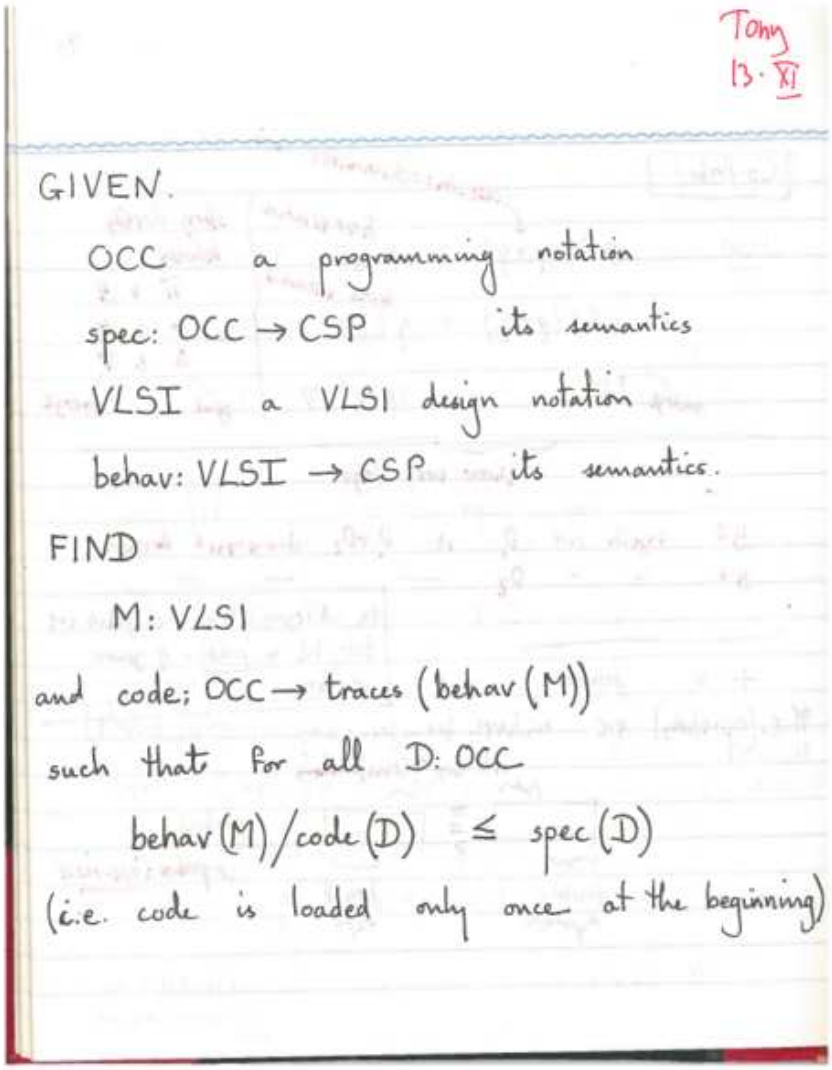
Hoare 87 — Le Pont d'Oye, Belgium "Friday the 13th" — 75

? Accountability.
· Machine must output current state on req.
Let tr = s^<req>^t^<qer> , ¬{req,qer} in t
then behav(M)/tr = behav(M)/s
≤ behav(M)/load(t)
where load = code∘dumpcrack.
FIND dumpcrack: dump → OCC.

*Paper read at the *BCS-FACS ProCoS Workshop on Provably Correct Systems*, London, UK, 9-10 March 2015.

As he wrote it, Tony carefully explained what he was after. To me that became the day of conception of the first ProCoS project.

I leave it to you to decipher the characteristic handwriting of Tony. OCC is some programming language. So is CSP. A specification maps programs in OCC into programs in CSP. VLSI is a language for specifying VLSI designs. Its semantics, in terms of CSP, is behav.



Tony
13. XI

GIVEN.

OCC a programming notation

spec: OCC → CSP its semantics

VLSI a VLSI design notation

behav: VLSI → CSP its semantics.

FIND

M: VLSI

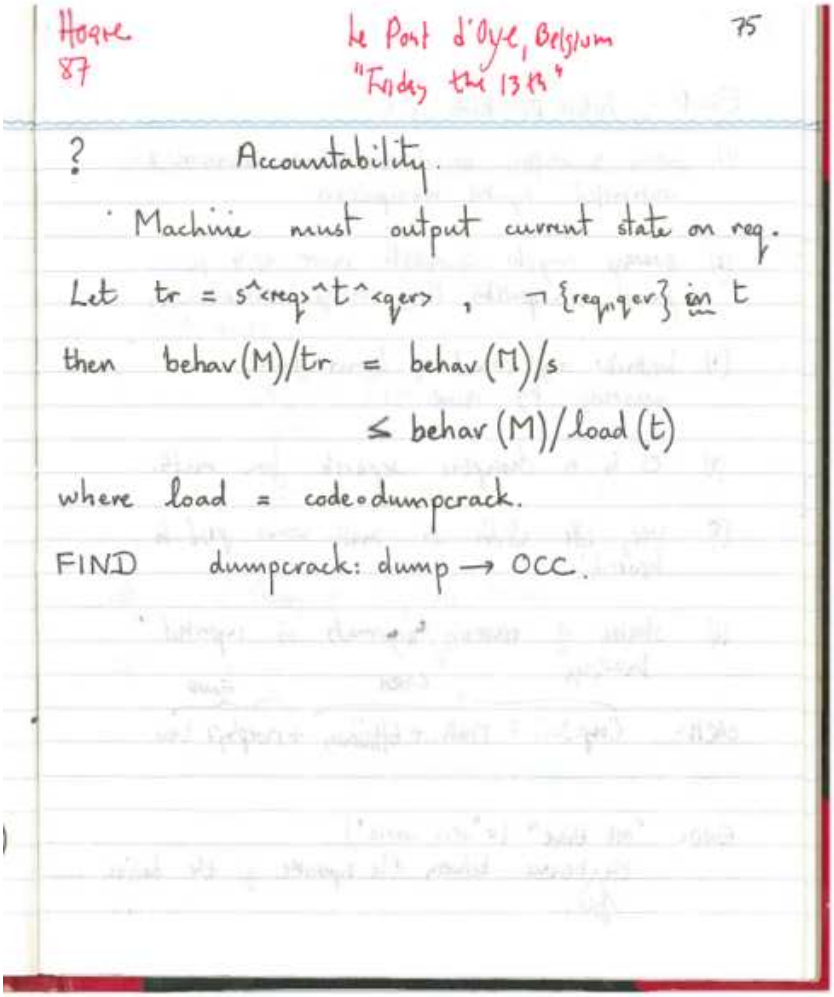
and code: OCC → traces (behav (M))

such that for all D: OCC

behav (M) / code (D) ≤ spec (D)

(i.e. code is loaded only once at the beginning)

Now find a VLSI implemented machine, M, and some OCC code which maps into traces of the behaviour of M such that for all programs, D, in OCC, the VLSI machine implements the OCC specification correctly.



Hoare 87 — Le Port d'Oye, Belgium "Friday the 13th" — 75

? Accountability.

· Machine must output current state on req.

Let tr = s^<req>^t^<qer> , ¬{req,qer} in t

then behav(M)/tr = behav(M)/s

≤ behav(M)/load(t)

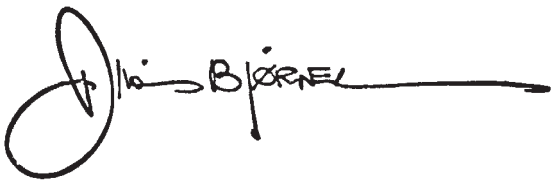
where load = code∘dumpcrack.

FIND dumpcrack: dump → OCC.

At the end of writing and narrating this, Tony asked: *should we try get an ESPRIT BRA project around this idea ?* We did.

• • •

To me a deciding moment of the project occurred during our Bornholm workshop. Prof. E.V. Sørensen had given a talk in which he sketched, from the background of his field, Circuit Theory, some ideas about handling digital signal transitions. I believe that Erling's talk gave impetus to the Duration Calculus. During the break, after EVS' talk, I saw Anders (Ravn), Tony and Zhou discussing, it appeared, the evolving DC ideas.



Holte, Denmark, October 30, 2015