From: **Guille Polito** guillermo.polito@inria.fr
Subject: Re: need your expert eyes
Date: 6 May 2020 at 23:37
To: Stéphane Ducasse stephane.ducasse@inria.fr
Cc: denker marcus.denker@inria.fr

GP

Ok, I read the blogpost and I gave a quick view on the “allocation” section of the pdf.

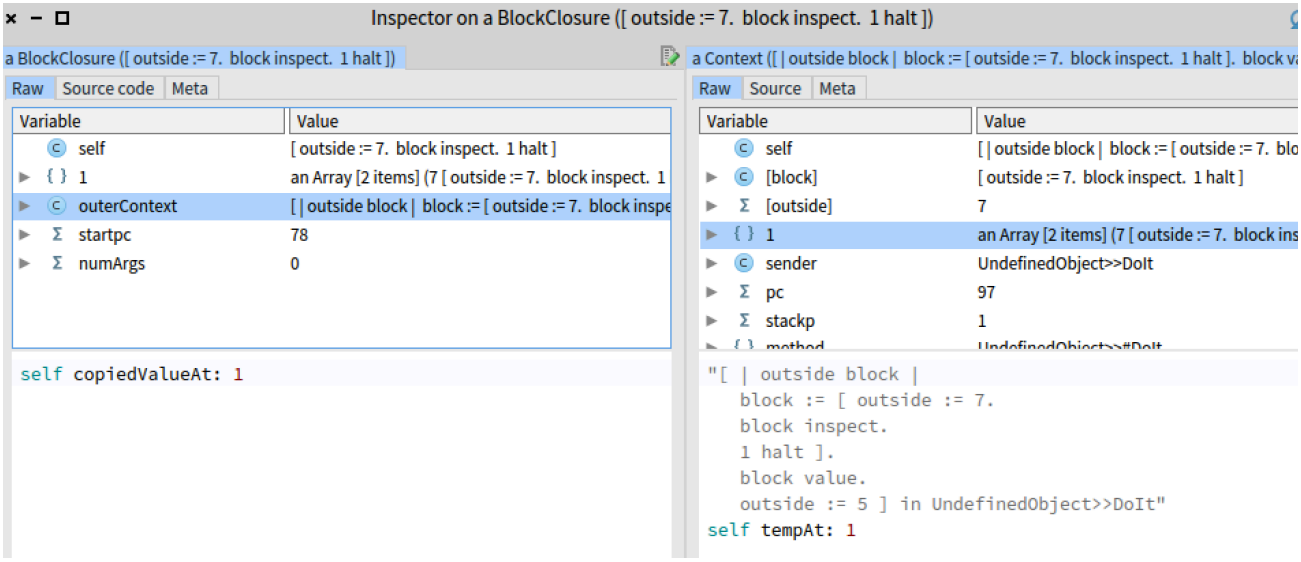
I think the post #2 can go as it is, as it talks conceptually about the lookup and so on.

I understand Marcus is referring to the “allocation” section on the pdf, because you asked explicitly to take a look at that :).
And I understand that this section looks pretty “implementation oriented” although it is not really faithful to the implementation we have.

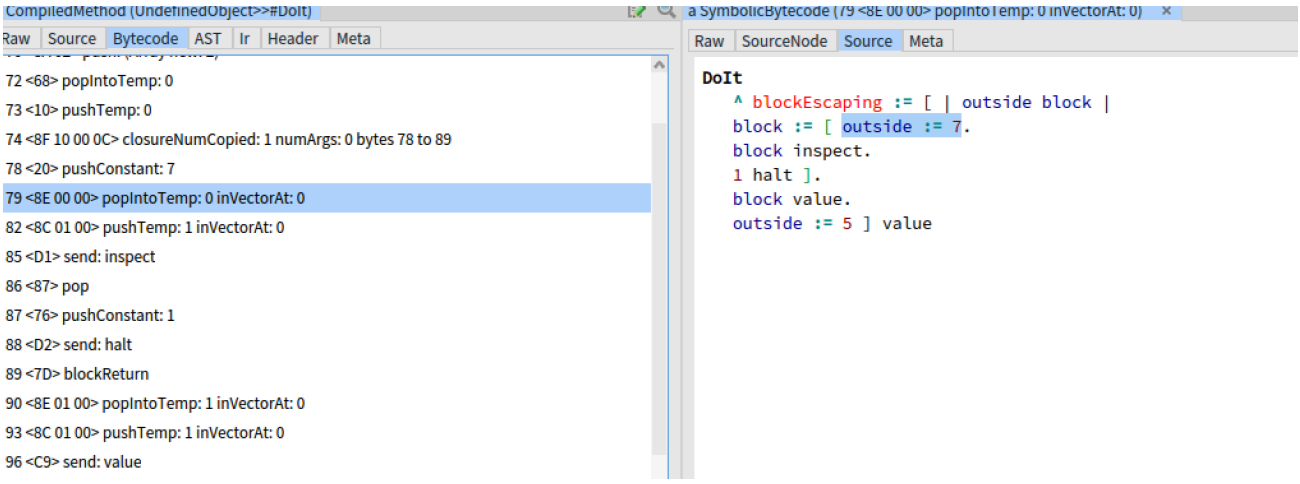
See for example:

```
[ | outside block |
    block := [
        outside := 7.
        block inspect.
        1halt. ].
    block value.
    outside := 5.
] value.
```

The inner block does actually have an array, which is an array shared with the home context ;)
And the value of the variable is in this array.
So at the end, there is no lookup.



Another way to see it is to look at how the code is compiled. The variables closed by the closure are actually read and written always by these pop*inVector push*inVector.



```
97 <87> pop
98 <23> pushConstant: 5
99 <8D 00 00> storeIntoTemp: 0 inVectorAt: 0
```

But then, it depends if in the booklet you want to focus on the conceptual model of blocks (in which all this is not necessary probably, because if you want to implement an AST interpreter you don't need this temp vector for example). Or if you want to give a more implementation-oriented "how it works our current implementation and how it is compiled".

> El 6 may 2020, a las 21:04, Guille Polito <guillermo.polito@inria.fr> escribió:
>
> > El 6 may 2020, a las 21:00, Stéphane Ducasse <stephane.ducasse@inria.fr> escribió:
> >
> > > I read it…
> > >
> > > I think it needs some work to explain the concept earlier that variables that are accessed from blocks are *not* accessed via the home context.
> >
> > Are you talking about the full block closures or also the closures we have in Pharo.
> >
> > > Right now the home context is used for a lot of explanations related to block variables, but with closures, they work completely independent of the home context.
> > >
> > > All vars that can be accessed in a closure are accessible locally, either because they are copied in (if they are only read) or because the array (temp vector) that they are stored in is copied in.
> >
> > Ok.
> > How can I see that?
> > Because I'm sure that this is not complex just nobody dared to explain it.
> >
> > I really think that we should do a better job at documenting this.
> >
> > > I need to think how to explain it better without getting too complex early on.
> >
> > Guille can you put on hold my blog post then.
>
> I think the temp vector is just a compilation detail and not inherent to the conceptual model lexical model of blocks. The conceptual model matches what you say: the lexical parent of a block is its home context.
>
> Maybe just a warning at the beginning should be enough.
>
> > S.
> >
> > > On 29 Apr 2020, at 15:02, Stéphane Ducasse <stephane.ducasse@inria.fr> wrote:
> > >
> > > Hi guys
> > >
> > > I did a first full version of the new block chapters and posts.
> > > And I would love your reading.
> > > I would like in particular you point on allocation.
> > >
> > > I removed the contents about context and I will go over it after I finish with the mooc migration.
> > >
> > > S.
> > >
> > > <Block.pdf>
> > >
> > > ----------
> > > Stéphane Ducasse

http://stephane.ducasse.free.fr / http://www.pharo.org
03 59 35 87 52
Assistant: Julie Jonas
FAX 03 59 57 78 50
TEL 03 59 35 86 16
S. Ducasse - Inria
40, avenue Halley,
Parc Scientifique de la Haute Borne, Bât.A, Park Plaza
Villeneuve d'Ascq 59650
France

Stéphane Ducasse
http://stephane.ducasse.free.fr / http://www.pharo.org
03 59 35 87 52
Assistant: Julie Jonas
FAX 03 59 57 78 50
TEL 03 59 35 86 16
S. Ducasse - Inria
40, avenue Halley,
Parc Scientifique de la Haute Borne, Bât.A, Park Plaza
Villeneuve d'Ascq 59650
France