



# eolang: L<sup>A</sup>T<sub>E</sub>X Package for Formulas and Graphs of EO Programming Language and $\varphi$ -calculus\*

Yegor Bugayenko  
yegor256@gmail.com

2022-10-30, 0.3.0

**NB!** You must run T<sub>E</sub>X processor with `--shell-escape` option and you must have [Perl](#) installed. This package doesn't work on Windows.

## 1 Introduction

This package helps you print formulas of  $\varphi$ -calculus, which is a formal foundation of [EO](#) programming language. The calculus was introduced by Bugayenko (2021) and later formalized by Kudasov et al. (2022). Here is how you render a simple expression:

|   |  |
|---|--|
| $  \begin{aligned}  a &\mapsto \llbracket \\  &\quad \rho \mapsto \xi.b, \\  &\quad b \mapsto \llbracket c \mapsto \text{fn}(56), \\  &\quad \quad \varphi \mapsto \text{hello}(\xi), \\  &\quad \quad \Delta \mapsto 01\text{-FE-C3} \rrbracket \rrbracket, \\  x &\mapsto \llbracket \alpha_0 \mapsto \emptyset \rrbracket.  \end{aligned}  $ | <pre> 1 \documentclass{article} 2 \pagestyle{empty} 3 \usepackage{eolang} 4 \begin{document} 5 \begin{phiquestion*} 6 a -&gt; [[ 7   ^ !-&gt; \$.b, 8   b -&gt; [[ c -&gt;  fn (56), 9     @ -&gt;  hello (\$), 10    \Delta ..&gt; 01-FE-C3 ]]]],\\ 11 x -&gt; [[ \alpha_0 -&gt; ? ]]. 12 \end{phiquestion*} 13 \end{document} </pre> |
|---|--|

`phiquestion (env)` The environment `phiquestion` lets you write a  $\varphi$ -calculus expressions using simple plain-text notation, where:

---

\*The sources are in GitHub at [objectionary/eolang.sty](https://github.com/objectionary/eolang.sty)

- “@” maps to “ $\varphi$ ” (`\varphi`),
- “^” maps to “ $\rho$ ” (`\rho`),
- “\$” maps to “ $\xi$ ” (`\xi`),
- “&” maps to “ $\sigma$ ” (`\sigma`),
- “?” maps to “ $\emptyset$ ” (`\varnothing`),
- “->” maps to “ $\mapsto$ ” (`\mapsto`),
- “~>” maps to “ $\rightsquigarrow$ ” (`\phiWave`),
- “!->” maps to “ $\vDash$ ” (`\phiConst`),
- “. .>” maps to “ $\vdash$ ” (`\phiDotted`),
- “[[” maps to “ $\llbracket$ ” (`\llbracket`),
- “]]” maps to “ $\rrbracket$ ” (`\rrbracket`),
- “|abc|” maps to “abc” (`\texttt{abc}`).

Also, a few symbols are supported for  $\varphi$ PU architecture:

- “-abc>” maps to “ $\xrightarrow{ABC}$ ” (`\xrightarrow{\text{\sffamily\scshape abc}}`),
- “:=” maps to “ $\models$ ” (`\vDash`).

`\phiiq` The command `\phiiq` lets you inline a  $\varphi$ -calculus expressions using the same simple plain-text notation:

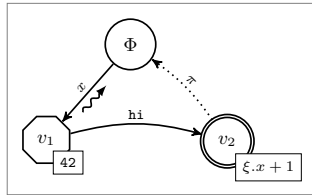
A simple object  $x \mapsto \llbracket \varphi \mapsto y \rrbracket$   
is a decorator of the data object  
 $y \mapsto \llbracket \Delta \mapsto 42 \rrbracket$ .

```

1 \documentclass{article}
2 \pagestyle{empty}
3 \usepackage{eolang}
4 \begin{document}
5 A simple object
6 \phiiq{x -> [[@ -> y]]} \
7 is a decorator of
8 the data object \
9 \phiiq{y -> [[\Delta .> 42]]}.
10 \end{document}

```

`sodg (env.)` The environment `sodg` allows you to draw a [SODG](#) graph:



```

1 \documentclass{article}
2 \pagestyle{empty}
3 \usepackage{eolang}
4 \begin{document}
5 \begin{sodg}
6 v0
7 v1 xy:v0,-2,+1 data:42
8 v0->v1 a:$x$ rho
9 v2 xy:v0,+1,+1 atom:\xi.x+1
10 v1->v2 a:|hi| bend:-15
11 v2->v0 pi bend:10
12 \end{sodg}
13 \end{document}

```

The content of the environment is parsed line by line. Markers in each line are separated by a single space. The first marker is either a unique name of a vertex, like  $v_1$  in the example above, or an edge, like  $v_0 \rightarrow v_1$ . All other markers are either unary like  $\rho$  or binary like  $\text{atom}:\$ \backslash x_i . x+1 \$$ . Binary markers have two parts, separated by colon. The following markers are supported for a vertex:

- “`data: [<box>]`” makes it a data vertex with an optional attached `<box>` (the content of the box may only be numeric data),
- “`atom: [<box>]`” makes it an atom with an optional attached `<box>` (the content of the box is a math formula),
- “`box:<txt>`” attaches a `<box>` to it,
- “`xy:<v>,<r>,<d>`” places this vertex in a position relative to the vertex `<v>`, shifting it right by `<r>` and down by `<d>` centimetres.

The following markers are supported for an edge:

- “`\rho`” places a backward snake arrow to the edge,
- “`\rrho`” places a reverse  $\rho$ ,
- “`bend:<angle>`” bend it right by the amount of `<angle>`,
- “`a:<txt>`” attaches label `<txt>` to it,
- “`\pi`” makes it dotted, with  $\pi$  label.

`\eolang` There is also a no-argument command `\eolang` to help you print the name of EO language. It understands anonymous mode of `\acmart` and prints itself differently, to `\xmir` double-blind your paper. There is also `\phic` command to print the name of  $\varphi$ -calculus, also sensitive to anonymous mode. The macro `\xmir` prints “XMIR”.

In our research we use XYZ,  
an experimental object-oriented  
dataflow language,  $\alpha$ -calculus,  
as its formal foundation, and XML’  
— its XML-based presentation.

```
1 \documentclass[anonymous]{acmart}
2 \thispagestyle{empty}
3 \usepackage{eolang}
4 \begin{document}
5 In our research we use \eolang{, \\\
6 an experimental object-oriented \\\
7 dataflow language, \phic{, \\\
8 as its formal foundation, and \xmir{ \\\
9 --- its XML-based presentation.
10 \end{document}
```

`\phiConst` A simple commands is defined to help you render an arrow for a constant attribute. It is recommende not to use it directly, but use `!->` instead. However, if you want to use `\phiConst`, wrap it in `\mathrel` for better display:

$$\llbracket x \mapsto y \rrbracket$$

```
1 \documentclass{article}
2 \pagestyle{empty}
3 \usepackage{eolang}
4 \begin{document}
5 \phiq{\llbracket x \mathrel{\phiConst} y \rrbracket}
6 \end{document}
```

## 2 Package Options

`tmpdir` The default location of temp files is `_eolang`. You can change this using `tmpdir` option:

```
\usepackage[tmpdir=/tmp/foo]{eolang}
```

## 3 More Examples

The `phiquation` environment treats ends of line as signals to start new lines in the formula. If you don't want this to happen and want to parse the next line as the a continuation of the current line, you can use a single backslash as it's done here:

|  |   |
|--|---|
| $\frac{x \mapsto [\varphi \mapsto y] \quad y \mapsto [z \mapsto 42]}{x.z \mapsto 42} R1$ | <pre> 1 \documentclass{article} 2 \usepackage{amsmath} 3 \usepackage{eolang} 4 \pagestyle{empty} 5 \begin{document} 6 \begin{phiquation*} 7 \dfrac \ 8 {x-&gt;[[@-&gt;y]] \quad y-&gt;[[z-&gt;42]]} \ 9 {x.z -&gt; 42} \ 10 \text{\sffamily R1} 11 \end{phiquation*} 12 \end{document} </pre> |
|--|---|

This is how you can use `\dfrac` from `amsmath` for large inference rules, with the help of `\begin{split}` and `\end{split}`:

|  |   |
|--|---|
| $\frac{x \mapsto [\varphi \mapsto y, z \mapsto 42, \alpha_0 \mapsto \emptyset, \alpha_1 \mapsto 42]}{x \mapsto [\varphi \mapsto y, z \mapsto \emptyset, f \rightsquigarrow \text{pi}(\alpha_0 \mapsto [\psi \mapsto \text{hello}(12)], \alpha_1 \mapsto 42)]} R2.$ | <pre> 1 \documentclass{article} 2 \usepackage{amsmath} 3 \usepackage{eolang} 4 \pagestyle{empty} 5 \begin{document} 6 \begin{phiquation*} 7 \dfrac{\begin{split} 8 x-&gt;[[@-&gt;y, z-&gt;42, 9 \quad \alpha_0-&gt;?, \alpha_1-&gt;42]] 10 \end{split}}{\begin{split} 11 x-&gt;[[@-&gt;y, z-&gt;?, f \rightsquigarrow  pi ( 12 \quad \alpha_0-&gt;[[\psi \mapsto  hello (12)]], 13 \quad \alpha_1-&gt;42]] 14 \end{split}} \text{R2}. 15 \end{phiquation*} 16 \end{document} </pre> |
|--|---|

The `phiquation` environment may be used together with [acmart](#):

$$\begin{array}{l}
x \mapsto \llbracket \\
\quad y \mapsto \llbracket \\
\quad \quad z \mapsto \xi, f \mapsto \emptyset \rrbracket, \\
\beta_1 \models [\psi \xrightarrow{\text{WAIT}} \emptyset].
\end{array}$$

```

1 \documentclass{acmart}
2 \usepackage{eolang}
3 \thispagestyle{empty}
4 \begin{document}
5 \begin{phiquestion*}
6 x -> [[
7   y -> [[
8     z !-> $, f ..> ? ]]]], \\
9 \beta_1 := [ \psi -wait> ? ].
10 \end{phiquestion*}
11 \end{document}

```

The phiquestion environment will automatically align formulas by the first arrow, if there are only left-aligned formulas:

$$\begin{array}{l}
x(\pi) \mapsto \llbracket \lambda \mapsto f_1 \rrbracket, \\
x(a, b, c) \mapsto \llbracket \alpha_0 \mapsto \emptyset, \varphi \mapsto \text{hello}(\xi) \rrbracket, \\
\Delta = 43-09.
\end{array}$$

```

1 \documentclass{acmart}
2 \usepackage{eolang}
3 \thispagestyle{empty}
4 \begin{document}
5 \begin{phiquestion*}
6 x(\pi) -> [[\lambda ..> f_1]], \\
7 x(a,b,c) -> [[ \alpha_0 -> ?, \
8   @ -> |hello|($) ]], \\
9 \Delta = |43-09|.
10 \end{phiquestion*}
11 \end{document}

```

## 4 Implementation

First, we include a few packages. We need [stmaryrd](#) for `\llbracket` and `\rrbracket` commands:

```
1 \RequirePackage{stmaryrd}
```

We need [amsmath](#) for `equation*` environment:

```
2 \RequirePackage{amsmath}
```

We need [amssymb](#) for `\varnothing` command. We disable `\Bbbk` because it may conflict with some packages from [acmart](#):

```
3 \let\Bbbk\relax\RequirePackage{amssymb}
```

We need [fancyvrb](#) for `\VerbatimEnvironment` command:

```
4 \RequirePackage{fancyvrb}
```

We need [iexec](#) for executing Perl scripts:

```
5 \RequirePackage{iexec}
```

Then, we process package options:

```

6 \RequirePackage{pgfopts}
7 \RequirePackage{ifluatex}
8 \RequirePackage{ifxetex}
9 \pgfkeys{

```

```

10 /eolang/.cd,
11 tmpdir/.store in=\eolang@tmpdir,
12 tmpdir/.default=_eolang\ifxetex-xe\else\ifluatex-lua\fi\fi,
13 tmpdir
14 }
15 \ProcessPgfoptions{/eolang}

```

Then, we make a directory where all temporary files will be kept:

```

16 \iexec[null]{mkdir -p "\eolang@tmpdir/\jobname"}%

```

\eolang@lineno Then, we define an internal counter to protect line number from changing:

```

17 \makeatletter\newcounter{eolang@lineno}\makeatother

```

\eolang@mdfive Then, we define a command for MD5 hash calculating of a file:

```

18 \RequirePackage{pdftexcmds}
19 \makeatletter
20 \newcommand\eolang@mdfive[1]{\pdf@filemdfivesum{#1}}
21 \makeatother

```

eolang-phi.pl Then, we create a Perl script for phiquation processing using VerbatimOut from [fancyvrb](#):

```

22 \makeatletter
23 \begin{VerbatimOut}{\eolang@tmpdir/eolang-phi.pl}
24 $env = $ARGV[0];
25 open(my $fh, '<', $ARGV[1]);
26 my $tex; { local $/; $tex = <$fh>; }
27 print '% This file is auto-generated', "\n";
28 print '% There are ', length($tex),
29 ' chars in the input: ', $ARGV[1], "\n";
30 print '% ---', "\n";
31 if (index($tex, "\t") > 0) {
32   print "TABS are prohibited!";
33   exit 1;
34 }
35 my @lines = split (/\\n/g, $tex);
36 foreach my $t (@lines) {
37   print '% ', $t, "\n";
38 }
39 print '% ---', "\n";
40 if ($env eq 'phiq') {
41   print '$';
42 } else {
43   print '\begin{', $env, '}\begin{split}';
44 }
45 $tex =~ s/^\s+|\s+$//g;
46 if ($env ne 'phiq') {
47   $tex =~ s/\\s+\\\\n\\s*//g;
48   $tex =~ s/\\\\\\\\n\\n\\n/g;
49 }
50 $tex =~ s/([\\s,>\\()([0-9A-F][0-9A-F-]*)/\\1|\\2/g;
51 $tex =~ s/\\/\\\\varnothing/g;
52 $tex =~ s/@/\\\\varphi/g;
53 $tex =~ s/&/\\\\sigma/g;
54 $tex =~ s/^/\\\\rho/g;

```

```

55 $tex =~ s/\$/\xi{/g;
56 $tex =~ s/-([a-z]+)>/\mathrel{\xrightarrow{\text{\sffamily\scshape \1}}}/g;
57 $tex =~ s/!->/\mathrel{\phiConst}/g;
58 $tex =~ s/->/\mathrel{\mapsto}/g;
59 $tex =~ s/~>/\mathrel{\phiWave}/g;
60 $tex =~ s/:=/\mathrel{\vDash}/g;
61 $tex =~ s/..>/\mathrel{\phiDotted}/g;
62 $tex =~ s/|([^\|]+)|/\textnormal{\texttt{\1}}{/g;
63 $tex =~ s/\[/\llbracket\mathrel{/g;
64 $tex =~ s/\]/\rrbracket\mathrel{/g;
65 if ($env ne 'phiq') {
66   $tex =~ s/\begin\{split\}\n/\begin{split}/g;
67   $tex =~ s/\n\s*\end\{split\}/\end{split}/g;
68   $tex =~ s/\n\n/\&/g;
69   $tex =~ s/\n\\\[-4pt]/g;
70   $tex =~ s/([^\&s])\s{2}([^\s])/ \1 \2/g;
71   $tex =~ s/\s{2}/ \quad/g;
72   my @leads = $tex =~ /&([^\s]+\s)/g;
73   my @eols = $tex =~ /\&/g;
74   $tex = '&' . $tex;
75   if (0+@leads == 0+@eols && 0+@eols > 0) {
76     $tex =~ s/&([^\s]+\s)/\1&/g;
77   }
78 }
79 print $tex;
80 if ($env eq 'phiq') {
81   print '$';
82 } else {
83   print '\end{split}\end{', $env, '>';
84 }
85 print '\endinput', "\n";
86 \end{VerbatimOut}
87 \message{eolang: File with Perl script
88   '\eolang@tmpdir/eolang-phi.pl' saved^^J}%
89 \iexec[trace,null]{perl -pi -e 's/(\\[a-zA-Z])\\s+/\1/g'
90   "\eolang@tmpdir/eolang-phi.pl"}
91 \makeatother

```

phiquation Then, we define phiquation and phiquation\* environments through a supplementary `\eolang@process` command:

```

92 \makeatletter\newcommand\eolang@process[1]{
93   \def\hash{\eolang@mdfive
94     {\eolang@tmpdir/\jobname/phiquation.tex}}%
95   \iexec[null]{cp "\eolang@tmpdir/\jobname/phiquation.tex"
96     "\eolang@tmpdir/\jobname/\hash.tex"}%
97   \iexec[trace,stdout=\eolang@tmpdir/\jobname/\hash-post.tex]{
98     perl "\eolang@tmpdir/eolang-phi.pl"
99     '#1'
100     "\eolang@tmpdir/\jobname/\hash.tex"}%
101   \setcounter{FancyVerbLine}{\value{eolang@lineno}}%
102 }
103 \newenvironment{phiquation*}%
104 {\VerbatimEnvironment%
105 \setcounter{eolang@lineno}{\value{FancyVerbLine}}%

```

```

106 \begin{VerbatimOut}
107   {\eolang@tmpdir/\jobname/phiuation.tex}}
108 {\end{VerbatimOut}\eolang@process{equation*}}
109 \newenvironment{phiuation}%
110 {\VerbatimEnvironment%
111 \setcounter{eolang@lineno}{\value{FancyVerbLine}}}%
112 \begin{VerbatimOut}
113   {\eolang@tmpdir/\jobname/phiuation.tex}}
114 {\end{VerbatimOut}\eolang@process{equation}}
115 \makeatother

```

`\phiiq` Then, we define `\phiiq` command:

```

116 \makeatletter\newcommand\phiiq[1]{%
117   \iexec[trace,quiet,stdout=\eolang@tmpdir/\jobname/phiq.tex]{
118     /bin/echo '\detokenize{#1}'}%
119   \def\hash{\eolang@mdfive
120     {\eolang@tmpdir/\jobname/phiq.tex}}%
121   \iexec[null]{cp "\eolang@tmpdir/\jobname/phiq.tex"
122     "\eolang@tmpdir/\jobname/\hash.tex"}%
123   \iexec[trace,stdout=\eolang@tmpdir/\jobname/\hash-post.tex]{
124     perl \eolang@tmpdir/eolang-phi.pl '\phiq'
125     "\eolang@tmpdir/\jobname/\hash.tex"}%
126 }\makeatother

```

`eolang-sodg.pl` Then, we create a Perl script for sodg graphs processing using `VerbatimOut` from [fancyvrb](#):

```

127 \makeatletter
128 \begin{VerbatimOut}{\eolang@tmpdir/eolang-sodg.pl}
129 open(my $fh, '<', $ARGV[0]);
130 my $tex; { local $/; $tex = <$fh>; }
131 print '% This file is auto-generated', "\n";
132 print '% There are ', length($tex),
133   ' chars in the input: ', $ARGV[0], "\n";
134 print '% ---', "\n";
135 if (index($tex, "\t") > 0) {
136   print "TABS are prohibited!";
137   exit 1;
138 }
139 $tex =~ s/^\s+|\s+$//g;
140 $tex =~ s/([a-zA-Z]+\s+)/\1/g;
141 $tex =~ s/\n\s+/\n/g;
142 $tex =~ s/|([^\|]+)|/\textnormal{\texttt{\1}}/g;
143 my @cmds = split (/\\n/g, $tex);
144 foreach my $t (@cmds) {
145   print '% ', $t, "\n";
146 }
147 print '% ---', "\n";
148 print '\begin{picture}', "\n";
149 foreach my $c (@cmds) {
150   my ($head, $tail) = split (/ /, $c, 2);
151   my %opts = {};
152   foreach my $p (split (/ /, $tail)) {
153     my ($q, $t) = split (/:/, $p);
154     $opts{$q} = $t;

```



```

155 }
156 if (index($head, '->') == -1) {
157     print '\node[';
158     if (exists $opts{'xy'}) {
159         my ($v, $right, $down) = split(/,/ , $opts{'xy'});
160         print ',below right=';
161         print $down;
162         print 'cm and ';
163         print $right;
164         print 'cm of ';
165         print $v;
166     }
167     if (exists $opts{'data'}) {
168         print ',phi-data';
169         if (not $opts{'data'} eq '') {
170             my $d = $opts{'data'};
171             if (index($d, '|') == -1) {
172                 $d = '\textnormal{\texttt{' . $d . '}}';
173             }
174             $opts{'box'} = $d;
175         }
176     } elsif (exists $opts{'atom'}) {
177         print ',phi-atom';
178         if (not $opts{'atom'} eq '') {
179             my $a = $opts{'atom'};
180             if (index($a, '$') == -1) {
181                 $a = '$' . $a . '$';
182             }
183             $opts{'box'} = $a;
184         }
185     } else {
186         print ',phi-object';
187     }
188     print ']';
189     print ' (' , $head, ')';
190     print ' {'$';
191     if ($head eq 'v0') {
192         print '\Phi';
193     } else {
194         print 'v_', substr($head, 1);
195     }
196     print '$}';
197     if (exists $opts{'box'}) {
198         print ' node[phi-box] at (';
199         print $head, '.south east) {';
200         print $opts{'box'}, '}';
201     }
202 } else {
203     print '\draw[';
204     if (exists $opts{'pi'}) {
205         print ',phi-pi';
206         if (not exists $opts{'a'}) {
207             $opts{'a'} = '$\pi$';
208         }

```

```

209 }
210 print ']';
211 my ($from, $to) = split (/>/, $head);
212 print '(', $from, ')';
213 if (exists $opts{'bend'}) {
214     print 'edge [bend right=', $opts{'bend'}, ']';
215 } else {
216     print '--';
217 }
218 if (exists $opts{'rho'} or exists $opts{'rrho'}) {
219     print ' pic[sloped,phi-rho]{parallel arrow={';
220     print '-' if not exists $opts{'rrho'};
221     print '0.3,-0.15}}';
222 }
223 if (exists $opts{'a'}) {
224     print ' node [phi-attr] {' , $opts{'a'}, '}';
225 }
226 print '(', $to, ')';
227 }
228 print ";\n";
229 }
230 print '\end{phicture}', "\n", '\endinput';
231 \end{VerbatimOut}
232 \message{eolang: File with Perl script
233   '\eolang@tmpdir/eolang-sodg.pl' saved^^J}%
234 \iexec[trace,null]{perl -pi -e 's/(\\[a-zA-Z])\\s+/\1/g'
235   "\eolang@tmpdir/eolang-sodg.pl"}
236 \makeatother

```

**FancyVerbLine** Then, we reset the counter for [fancyvrb](#), so that it starts counting lines from zero when the document starts rendering:

```

237 \setcounter{FancyVerbLine}{0}

```

**tikz** Then, we include tikz package and its libraries:

```

238 \RequirePackage{tikz}
239 \usetikzlibrary{arrows}
240 \usetikzlibrary{shapes}
241 \usetikzlibrary{decorations}
242 \usetikzlibrary{decorations.pathmorphing}
243 \usetikzlibrary{intersections}
244 \usetikzlibrary{positioning}
245 \usetikzlibrary{calc}
246 \usetikzlibrary{shapes.arrows}

```

**phicture** Then, we define internal environment phicture:

```

247 \newenvironment{phicture}%
248   {\noindent\begin{tikzpicture}[
249     ->,>=stealth',node distance=0,thick,
250     pics/parallel arrow/.style={
251       code={\draw[-latex,phi-rho] (##1) -- (-##1);}}}%
252   {\end{tikzpicture}}
253 \tikzstyle{transforms} = [fill=white!80!black, single arrow,
254   minimum height=0.5cm, minimum width=0.5cm,
255   single arrow head extend=2mm]

```

```

256 \tikzstyle{phi-thing} = [thick,inner sep=0pt,minimum height=2.4em,
257   draw,font={\small}]
258 \tikzstyle{phi-object} = [phi-thing,circle]
259 \tikzstyle{phi-data} = [phi-thing,regular polygon,
260   regular polygon sides=8]
261 \tikzstyle{phi-empty} = [phi-object]
262 \tikzstyle{phi-rho} = [draw,decorate,decoration={
263   snake,amplitude=.4mm,segment length=2mm,post length=1mm}]
264 \tikzstyle{phi-pi} = [draw,dotted]
265 \tikzstyle{phi-atom} = [phi-object,double]
266 \tikzstyle{phi-box} = [xshift=-5pt,yshift=3pt,draw,fill=white,
267   rectangle,thin,minimum width=1.2em,anchor=north west,
268   font={\scriptsize}]
269 \tikzstyle{phi-attr} = [midway,sloped,inner sep=0pt,
270   above=2pt,sloped/.append style={transform shape},
271   font={\scriptsize},color=black]

```

sodg Then, create a new environment sodg, as suggested [here](#):

```

272 \makeatletter\newenvironment{sodg}%
273 {\VerbatimEnvironment%
274 \setcounter{eolang@lineno}{\value{FancyVerbLine}}%
275 \begin{VerbatimOut}
276   {\eolang@tmpdir/\jobname/sodg.tex}}
277 {\end{VerbatimOut}}%
278 \def\hash{\eolang@mdfive
279   {\eolang@tmpdir/\jobname/sodg.tex}}%
280 \iexec[null]{cp "\eolang@tmpdir/\jobname/sodg.tex"
281   "\eolang@tmpdir/\jobname/\hash.tex"}%
282 \iexec[trace,stdout=\eolang@tmpdir/\jobname/\hash-post.tex]{
283   perl "\eolang@tmpdir/eolang-sodg.pl"
284   "\eolang@tmpdir/\jobname/\hash.tex"}%
285 \setcounter{FancyVerbLine}{\value{eolang@lineno}}%
286 }\makeatother

```

\eolang Then, we define a simple supplementary command to help you print EO, the name of our language.

```

287 \newcommand\eolang{%
288   \ifdefined\anon%
289     \anon[XYZ]{\sffamily EO}}%
290   \else%
291     {\sffamily EO}%
292   \fi%
293 }

```

\phic Then, we define a simple supplementary command to help you print  $\varphi$ -calculus, the name of our formal apparatus.

```

294 \RequirePackage{hyperref}
295 \newcommand\phic{%
296   \ifdefined\anon%
297     \anon[\texorpdfstring{\alpha}{a}-calculus]{\texorpdfstring{\varphi}{\phi}-calculus}%
298   \else%
299     \texorpdfstring{\varphi}{\phi}-calculus%
300   \fi%
301 }

```

`\xmir` Then, we define a simple supplementary command to help you print XMIR, the name of our XML-based format of program representation.

```
302 \newcommand\xmir{%
303   \ifdefined\anon%
304     \anon[XML']{XMIR}%
305   \else%
306     XMIR%
307   \fi%
308 }
```

`\phiConst` Then, we define a command to render an arrow for a constant attribute, as suggested [here](#):

```
309 \newcommand\phiConst{%
310   \mathrel{\hspace{.15em}}\mapstochar\mathrel{\hspace{-.15em}}\mapsto}
```

`\phiWave` Then, we define a command to render an arrow for a multi-layer attribute, as suggested [here](#):

```
311 \newcommand\phiWave{%
312   \mapstochar\mathrel{\mspace{0.45mu}}\leadsto}
```

`\phiDotted` Then, we define a command to render an arrow for a special attribute, as suggested [here](#):

```
313 \RequirePackage{trimclip}
314 \RequirePackage{amsfonts}
315 \makeatletter
316 \newcommand{\phiDotted}{\mapstochar\mathrel{\mathpalette\phiDotted@\relax}}
317 \newcommand{\phiDotted@}[2]{%
318   \begingroup
319     \settowidth{\dimen\z@}{\m@th#1\rightarrow$}%
320     \settoheight{\dimen\tw@}{\m@th#1\rightarrow$}%
321     \sbox\z@{%
322       \makebox[\dimen\z@][s]{%
323         \clipbox{0 0 {0.4\width} 0}%
324         {\resizebox{\dimen\z@}{\height}%
325           {\m@th#1\dashrightarrow$}}%
326         \hss%
327         \clipbox{{0.69\width} {-0.1\height} 0 {-\height}}{\m@th#1\rightarrow$}%
328       }%
329     }%
330     \ht\z@=\dimen\tw@ \dp\z@=\z@%
331     \box\z@%
332   \endgroup\makeatother}
```

## References

- Bugayenko, Yegor (2021). *EOLANG and  $\varphi$ -calculus*. arXiv: [2111.13384](#) [cs.PL].
- Kudasov, Nikolai et al. (2022).  *$\varphi$ -calculus: a purely object-oriented calculus of decorated objects*. arXiv: [2204.07454](#) [cs.PL].

## Change History

|  |  |   |
|--|--|---|
| 0.0.1  |  |   |
| General: First draft. . . . .  | 5  |   |
| 0.0.2  |  |   |
| sodg: The environment “phigure”<br>renamed to “sodg” for the sake of<br>better semantic. The graph in the<br>picture is solely a SODG graph,<br>that’s why the name “sodg” is<br>better. . . . . | 11   |   |
| eolang-phi.pl: New symbol added<br>for basket slots . . . . .  | 6  |   |
| Parsing of symbols “@,” “^,” and “&”<br>enabled (varphi, rho, and sigma) . . .   | 6  |   |
| The symbols “[” and “]” replaced<br>with “[[” and “]]” for abstract<br>object brackets, because they<br>conflicted with normal square<br>brackets . . . . .                                      | 6  |   |
| eolang-sodg.pl: The Perl file now<br>has a fixed name, which doesn’t<br>depend on the name of the TeX job.<br>This file may be shared among jobs,<br>no need to make it uniquely named. . .      | 8  |   |
| \phiq: Parsing of additional symbols<br>enabled. . . . .   | 8  |   |
| 0.1.0  |  |   |
| General: Parsing of package options<br>introduced. . . . .   | 5  |   |
| \eolang: New command “eolang”<br>added to print the name of the<br>language in both normal and<br>anonymous mode of “acmart” . . .   | 11   |   |
| \eolang@mdfive: New<br>supplementary command added to<br>calculate MD5 sum of a file. . . . .  | 6  |   |
|  |  | eolang-phi.pl: A new Perl script<br>“eolang-phi.pl” added for parsing of<br>phi expressions. . . . .                              |
|  |  | 6   |
|  |  | eolang-sodg.pl: There are two Perl<br>scripts now: one for phiquation,<br>another one for sodg. . . . .                           |
|  |  | 8   |
|  |  | \phic: New command “phic” prints<br>the name of $\varphi$ -calculus in both<br>normal and anonymous mode of<br>“acmart” . . . . . |
|  |  | 11  |
|  |  | \phiConst: New command “phiConst”<br>added to denote a link to a<br>constant attribute. . . . .                                   |
|  |  | 12  |
|  |  | \phiDotted: New command<br>“phiDotted” added to denote a link<br>to a special attribute. . . . .                                  |
|  |  | 12  |
|  | 0.2.0  |   |
|  | eolang-phi.pl: Numbers<br>automatically render as “texttt”. No<br>need to use vertical bars around<br>them anymore. . . . .            | 6   |
|  | eolang-sodg.pl: The content of<br>“atom” and “data” boxes is parsed<br>automatically as formulas and<br>numbers, respectively. . . . . | 8   |
|  | \xmirl: New command “xmirl” prints<br>XMIR in both normal and<br>anonymous mode of “acmart” . . .                                      | 12  |
|  | 0.3.0  |   |
|  | \eolang@lineno: New counter for<br>protecting lineno. . . . .  | 6   |
|  | eolang-phi.pl: New arrow added,<br>that looks like “leadsto”. . . . .  | 6   |
|  | \phiWave: New command “phiWave”<br>added to denote a link to a<br>multi-layer attribute. . . . .                                       | 12  |

# Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

|   |                         |  |
|---|-------------------------|--|
| <b>Symbols</b>                                    |                         |  |
| $\backslash \$$ .....                             | 55                      |  |
| $\backslash ($ .....                              | 50                      |  |
| $\backslash ?$ .....                              | 51                      |  |
| $\backslash [$ .....                              | 63                      |  |
| $\backslash \{$ .....                             | 66, 67                  |  |
| $\backslash \}$ .....                             | 66, 67                  |  |
| $\backslash ]$ .....                              | 64                      |  |
| $\backslash ^$ .....                              | 54                      |  |
| $\backslash  $ .....                              | 62, 142                 |  |
| <b>Numbers</b>                                    |                         |  |
| $\backslash 2$ .....                              | 50, 70                  |  |
| <b>A</b>  |                         |  |
| $\backslash \alpha$ .....                         | 297                     |  |
| $\backslash \text{anon}$ .....                    | 288,                    |  |
|   | 289, 296, 297, 303, 304 |  |
| <b>B</b>  |                         |  |
| $\backslash \text{Bbbk}$ .....                    | 3                       |  |
| $\backslash \text{begin}$ ....                    | 23, 43, 106,            |  |
|   | 112, 128, 148, 248, 275 |  |
| $\backslash \text{box}$ .....                     | 331                     |  |
| <b>C</b>  |                         |  |
| $\backslash \text{clipbox}$ .....                 | 323, 327                |  |
| <b>D</b>  |                         |  |
| $\backslash \text{dashrightarrow}$ ...            | 325                     |  |
| $\backslash \text{def}$ .....                     | 93, 119, 278            |  |
| $\backslash \text{detokenize}$ .....              | 118                     |  |
| $\backslash \text{dimen}$ 319, 320, 322, 324, 330 |                         |  |
| $\backslash \text{dp}$ .....                      | 330                     |  |
| $\backslash \text{draw}$ .....                    | 203, 251                |  |
| <b>E</b>  |                         |  |
| $\backslash \text{end}$ .....                     | 83, 86, 108,            |  |
|   | 114, 230, 231, 252, 277 |  |
| $\backslash \text{endinput}$ .....                | 85, 230                 |  |
| $\backslash \text{eolang}$ .....                  | 287                     |  |
| $\backslash \text{eolang-phi.pl}$ .....           | 22                      |  |
| $\backslash \text{eolang-sodg.pl}$ ...            | 127                     |  |
| $\backslash \text{eolang@lineno}$ .....           | 17                      |  |
| $\backslash \text{eolang@mdfive}$ ....            |                         |  |
|   | .... 18, 93, 119, 278   |  |
| $\backslash \text{eolang@process}$ ...            |                         |  |
|   | .... 92, 108, 114       |  |
| $\backslash \text{eolang@tmpdir}$ ....            |                         |  |
|   | .... 11, 16, 23, 88,    |  |
|   | 90, 94, 95, 96, 97,     |  |
|   | 98, 100, 107, 113,      |  |
|   | 117, 120, 121, 122,     |  |
|   | 123, 124, 125, 128,     |  |
|   | 233, 235, 276, 279,     |  |
|   | 280, 281, 282, 283, 284 |  |
| <b>F</b>  |                         |  |
| $\backslash \text{FancyVerbLine}$ ....            | 237                     |  |
| <b>H</b>  |                         |  |
| $\backslash \text{hash}$ .....                    | 93, 96, 97,             |  |
|   | 100, 119, 122, 123,     |  |
|   | 125, 278, 281, 282, 284 |  |
| $\backslash \text{height}$ .....                  | 324, 327                |  |
| $\backslash \text{hspace}$ .....                  | 310                     |  |
| $\backslash \text{hss}$ .....                     | 326                     |  |
| $\backslash \text{ht}$ .....                      | 330                     |  |
| <b>I</b>  |                         |  |
| $\backslash \text{iexec}$ 16, 89, 95, 97, 117,    |                         |  |
|   | 121, 123, 234, 280, 282 |  |
| $\backslash \text{ifdefined}$ ..                  | 288, 296, 303           |  |
| $\backslash \text{ifluatex}$ .....                | 12                      |  |
| $\backslash \text{ifxetex}$ .....                 | 12                      |  |
| <b>J</b>  |                         |  |
| $\backslash \text{jobname}$ .....                 | 16, 94,                 |  |
|   | 95, 96, 97, 100, 107,   |  |
|   | 113, 117, 120, 121,     |  |
|   | 122, 123, 125, 276,     |  |
|   | 279, 280, 281, 282, 284 |  |
| <b>L</b>  |                         |  |
| $\backslash \text{leadsto}$ .....                 | 312                     |  |
| <b>M</b>  |                         |  |
| $\backslash \text{m@th}$ ...                      | 319, 320, 325, 327      |  |
| $\backslash \text{makeatletter}$ 17, 19, 22,      |                         |  |
|   | 92, 116, 127, 272, 315  |  |
| $\backslash \text{makeatother}$ 17, 21, 91,       |                         |  |
|   | 115, 126, 236, 286, 332 |  |
| $\backslash \text{makebox}$ .....                 | 322                     |  |
| $\backslash \text{mapsto}$ .....                  | 310                     |  |
| $\backslash \text{mapstochar}$ .                  | 310, 312, 316           |  |
| $\backslash \text{mathpalette}$ .....             | 316                     |  |
| $\backslash \text{mathrel}$ ...                   | 310, 312, 316           |  |
| $\backslash \text{message}$ .....                 | 87, 232                 |  |
| $\backslash \text{mspace}$ .....                  | 312                     |  |
| <b>N</b>  |                         |  |
| $\backslash \text{newcommand}$ .....              | 20,                     |  |
|   | 92, 116, 287, 295,      |  |
|   | 302, 309, 311, 316, 317 |  |
| $\backslash \text{newcounter}$ .....              | 17                      |  |
| $\backslash \text{newenvironment}$ ...            |                         |  |
|   | .... 103, 109, 247, 272 |  |
| $\backslash \text{node}$ .....                    | 157                     |  |
| $\backslash \text{noindent}$ .....                | 248                     |  |
| <b>P</b>  |                         |  |
| $\backslash \text{pdf@filemdfivesum}$ ..          | 20                      |  |
| $\backslash \text{pgfkeys}$ .....                 | 9                       |  |
| $\backslash \Phi$ .....                           | 192                     |  |
| $\backslash \text{phic}$ .....                    | 294                     |  |
| $\backslash \text{phiConst}$ .....                | 309                     |  |
| $\backslash \text{picture}$ .....                 | 247                     |  |
| $\backslash \text{phiDotted}$ .....               | 313                     |  |
| $\backslash \text{phiDotted@}$ ....               | 316, 317                |  |
| $\backslash \text{phiq}$ .....                    | 116                     |  |
| $\backslash \text{phiquation}$ .....              | 92                      |  |
| $\backslash \text{phiWave}$ .....                 | 311                     |  |
| $\backslash \pi$ .....                            | 207                     |  |
| $\backslash \text{ProcessPgfoptions}$ ..          | 15                      |  |
| <b>R</b>  |                         |  |
| $\backslash \text{relax}$ .....                   | 3, 316                  |  |
| $\backslash \text{RequirePackage}$ ..             | 1,                      |  |
|   | 2, 3, 4, 5, 6, 7, 8,    |  |
|   | 18, 238, 294, 313, 314  |  |
| $\backslash \text{resizebox}$ .....               | 324                     |  |
| $\backslash \text{rightarrow}$ .                  | 319, 320, 327           |  |
| <b>S</b>  |                         |  |
| $\backslash \text{sbox}$ .....                    | 321                     |  |
| $\backslash \text{scriptsize}$ ....               | 268, 271                |  |
| $\backslash \text{setcounter}$ ....               | 101,                    |  |
|   | 105, 111, 237, 274, 285 |  |
| $\backslash \text{settoheight}$ .....             | 320                     |  |
| $\backslash \text{settowidth}$ .....              | 319                     |  |
| $\backslash \text{sffamily}$ .....                | 289, 291                |  |
| $\backslash \text{small}$ .....                   | 257                     |  |
| $\backslash \text{sodg}$ .....                    | 272                     |  |
| <b>T</b>  |                         |  |
| $\backslash \text{t}$ .....                       | 31, 135                 |  |

|                                    |          |   |   |
|------------------------------------|----------|---|---|
| <code>\texorpdfstring</code>       | 297, 299 | <b>U</b>                                    | <b>W</b>                                      |
| <code>\textnormal</code> . . . . . | 172      | <code>\usetikzlibrary</code> . . .          | <code>\width</code> . . . . . 323, 327        |
| <code>\texttt</code> . . . . .     | 172      | . . . 239, 240, 241,                        |   |
| <code>\tikz</code> . . . . .       | 238      | 242, 243, 244, 245, 246                     | <b>X</b>                                      |
| <code>\tikzstyle</code> . . . . .  | 253,     | <b>V</b>                                    | <code>\xmir</code> . . . . . 302              |
| 256, 258, 259, 261,                |          | <code>\value</code> 101, 105, 111, 274, 285 |   |
| 262, 264, 265, 266, 269            |          | <code>\varphi</code> . . . . . 297, 299     | <b>Z</b>                                      |
| <code>\tw@</code> . . . . .        | 320, 330 | <code>\VerbatimEnvironment</code>           | <code>\z@</code> 319, 321, 322, 324, 330, 331 |
|                                    |          | . . . . . 104, 110, 273                     |   |