Lambda Notation

by Sven Nilsen, 2018

A normal mathematical function[1] can be written like this:

```
f(x) = x + 1
```

Sometimes it is desired to treat functions anonymuously, called a “lambda”, “closure”.

In lambda calculus[2], it is common to use this notation:

```
f := \x.x+1
```

In path semantics[3], the notation for writing lambdas is the following:

```
f := \(x) = x + 1
```

This follows the notation used in the Dyon programming language[4].

I prefer to use this in path semantics[3] for the following reasons:

- The notation is more familiar for programmers
- The notation is closer to the normal mathematical form[1] (replacying name by “\”)
- It is easier to make lambdas more explicit, since in some cases,
  function currying[5] is not permitted, where multiple arguments are needed

References:

[1] “Function (mathematics)”
Wikipedia
https://en.wikipedia.org/wiki/Function_(mathematics)

[2] “Lambda calculus”
Wikipedia
https://en.wikipedia.org/wiki/Lambda_calculus

[3] “Path semantics”
AdvancedResearch, Sven Nilsen
https://github.com/advancedresearch/path_semantics

[4] “Dyon programming language”
The Piston Project, Sven Nilsen
https://github.com/pistondevelopers/dyon

[5] “Currying”
Wikipedia
https://en.wikipedia.org/wiki/Currying