

Semantic Theory 2017: Installing PDRT-SANDBOX

In the following weeks, we are going to use PDRT-SANDBOX, a Haskell library that implements Discourse Representation Theory (DRT), and its extension Projective Discourse Representation Theory (PDRT).

The goal of this tutorial is to download and install PDRT-SANDBOX, and to familiarize yourselves with it.

Installing PDRT-SANDBOX

PDRT-SANDBOX is written in Haskell (for reasons that will soon become clear), and as such you will need a Haskell compiler and interpreter to run it. The easiest way to get a Haskell compiler and interpreter is to install **The Haskell Platform** which runs on Windows, Mac, and Linux, and is available at:

```
http://www.haskell.org/platform/
```

To learn about basic Haskell commands, see the very great introduction to Haskell available at:

```
http://learnyouahaskell.com/
```

Once you have installed **The Haskell Platform**, you can download and install PDRT-SANDBOX, which is available at:

```
http://hbrouwer.github.io/pdrt-sandbox/
```

In the source directory, type:

```
$ make
```

If you do not have `make`, try:

```
$ runhaskell Setup.hs configure --prefix=${HOME} --user  
$ runhaskell Setup.hs build  
$ runhaskell Setup.hs install
```

If all went well, you should now be able to use PDRT-SANDBOX, by importing it in the Haskell interpreter `ghci`:

```
$ ghci
GHCi, version 7.8.3: http://www.haskell.org/ghc/  :? for help
Loading package ghc-prim ... linking ... done.
Loading package integer-gmp ... linking ... done.
Loading package base ... linking ... done.
Prelude> :m Data.DRS
Prelude Data.DRS>
```

PDRT-SANDBOX Tutorial

Now that you are all set up, familiarize yourselves with PDRT-SANDBOX by doing the tutorial `DRSTutorial.hs` (which comes with the source bundle, but is also available from the PDRT-SANDBOX website). You can easily try the examples in this file by loading the file in `ghci`:

```
$ ghci DRSTutorial.hs
```

Try to define some new DRSs yourself using the PDRT-SANDBOX syntax. Perform merge operations on these DRSs and try to define some of the lambda-DRSs that were used in the slides. Experiment with it!