Programming Languages 2019-1: Problem Set 1

Luis Daniel Aragon Bermudez   Facultad de Ciencias, UNAM

August 17th, 2018

Contents

Introduction

Two modules are provided in this repository as solutions for the problem set:

- `ListS` contains a Snoc lists representation with a few useful functions built around it.
- `Nat` contains an ambiguous (see conclusions) natural numbers representation that uses three useful constructors and a few functions built around it.

Using the program

Lists

This module can be loaded on the interactive shell by issuing the following command from the project's root directory:

```
$ ghci src/ListS.hs
```

`headS`: This function retrieves the first element of the list.

```
*ListS> headS NilS
*** Exception: Empty list
*ListS> headS (Snoc NilS 1 )
1
*ListS> headS (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
1
```

`tailS`: This function retrieves the list without the first element.

```
*ListS> tailS NilS
*** Exception: Empty list
*ListS> tailS (Snoc NilS 1 )
NilS
*ListS> tailS (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc NilS 2) 3) 4) 5
```

`initS`: This function retrieves the list without the last element.

```
*ListS> initS NilS
*** Exception: Empty list
*ListS> initS (Snoc NilS 1 )
NilS
*ListS> initS (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc NilS 1) 2) 3) 4
```

`lastS`: This function retrieves the last element of the list.

```
*ListS> lastS NilS
*** Exception: Empty list
*ListS> lastS (Snoc NilS 1 )
1
*ListS> lastS (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
5
```

`nthElementS`: This function retrieves the n-th element of the list.

```
*ListS> nthElementS 5 NilS
*** Exception: Invalid index
*ListS> nthElementS 10 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
*** Exception: Invalid index
*ListS> nthElementS (-1) (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
*** Exception: Invalid index
*ListS> nthElementS 0 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
1
*ListS> nthElementS 2 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
3
```

`deleteNthElementS`: This function retrieves the list without its n-th element.

```
*ListS> deleteNthElementS 5 NilS
NilS
*ListS> deleteNthElementS 10 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc (Snoc NilS 1) 2) 3) 4) 5
*ListS> deleteNthElementS (-1) (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
*** Exception: Invalid index
*ListS> deleteNthElementS 0 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc NilS 2) 3) 4) 5
*ListS> deleteNthElementS 2 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc NilS 1) 2) 4) 5
```

`addFirstS`: Given an element ‘a’ and a list ‘x’, this function retrieves the list ‘ax’.

```
*ListS> addFirstS 1 NilS
Snoc NilS 1
*ListS> addFirstS 2 (Snoc NilS 1 )
Snoc (Snoc NilS 2) 1
*ListS> addFirstS 6 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc (Snoc (Snoc NilS 6) 1) 2) 3) 4) 5
```

`addLastS`: Given an element ‘a’ and a list ‘x’, this function retrieves the list ‘xa’

```
*ListS> addLastS 1 NilS
Snoc NilS 1
*ListS> addLastS 2 (Snoc NilS 1 )
Snoc (Snoc NilS 1) 2
*ListS> addLastS 6 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1) 2) 3) 4) 5) 6
```

`reverseS`: This function reverses a Snoc list.

```
*ListS> reverseS NilS
NilS
*ListS> reverseS (Snoc NilS 1 )
Snoc NilS 1
*ListS> reverseS (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc (Snoc (Snoc (Snoc NilS 5) 4) 3) 2) 1
```

`appendS`: This function concatenates two snoc lists.

```
*ListS> appendS NilS (Snoc NilS 1 )
Snoc NilS 1
*ListS> appendS (Snoc (Snoc (Snoc NilS 1 ) 2) 3) (Snoc (Snoc NilS 4) 5)
Snoc (Snoc (Snoc (Snoc (Snoc NilS 1) 2) 3) 4) 5
```

`takeS`: This function retrieves the list's 'n' first elements.

```
*ListS> takeS 1 NilS
NilS
*ListS> takeS 2 (Snoc NilS 1 )
Snoc NilS 1
*ListS> takeS 2 (Snoc (Snoc (Snoc (Snoc (Snoc NilS 1 ) 2) 3) 4) 5)
Snoc (Snoc NilS 1) 2
```

Naturals

This module can be loaded on the interactive shell by issuing the following command from the project's root directory:

```
$ ghci src/Nat.hs
```

`toNat`: Retrieves an Int's Nat equivalent.

```
*Nat> toNat 0
Zero
*Nat> toNat 1
O Zero
*Nat> toNat 2
D (O Zero)
```

`succN`: Retrieves the successor of a Nat.

```
*Nat> succN Zero
O Zero
*Nat> succN (O Zero))
D (O Zero)
*Nat> succN (D (O Zero))
O (O Zero)
```

`predN`: Retrieves the predecessor of a Nat.

```
*Nat> predN (O Zero))
Zero
*Nat> predN (D (O Zero))
O Zero
```

`addN`: Retrieves the addition of two Nat's.

```
*Nat> addN Zero (O Zero)
O Zero
*Nat> addN (O Zero) (D (O Zero))
O (O Zero)
```

```
*Nat> addN (D (O Zero)) (D (O Zero))
D (D (O Zero))
```

`prodN`: Retrieves the product of two Nat's.

```
*Nat> prodN Zero (O Zero)
Zero
*Nat> prodN (O Zero) (D (O Zero))
D (O Zero)
*Nat> prodN (D (O Zero)) (D (O Zero))
D (D (O Zero))
```

Conclusions

Lists

Many operations are deeply recursive, thus this isn't a performance-first implementation. Many improvements could be done at the cost of some semantic simplicity.

Naturals

The Nat type is not a good spec for natural numbers because 0 has potentially infinite representations and a zero element should be unique (ProofWiki contributors 2013). In order for the functions to work properly, we assume developers will not use expressions like `D (D (D ... (D Zero) ... ))` to represent 0.

Acknowledgements

This project was developed as part of the activities of the 2019 Programming Languages[1] course taught by Prof. Favio Ezequiel Miranda Perea PhD at the School of Sciences of the National Autonomous University of Mexico[2].

This readme was created using pandoc[3]. This program is written in haskell[4], a great functional language.

Bibliography

"Pandoc User's Guide." n.d. *Pandoc - About Pandoc*. https://pandoc.org/MANUAL.html.

ProofWiki contributors. 2013. "Zero Element Is Unique." *ProofWiki*. ProofWiki. https://proofwiki.org/wiki/Zero_Element_is_Unique.

Wikipedia contributors. 2018. "Haskell (Programming Language) — Wikipedia, the Free Encyclopedia." https://en.wikipedia.org/w/index.php?title=Haskell_(programming_language)&oldid=854056702.

[1]Course's site can be found here.
[2]Facultad de Ciencias, UNAM.
[3]Pandoc is a Haskell library for converting from one markup format to another ("Pandoc User's Guide," n.d.).
[4]Haskell is a standardized, general-purpose compiled purely functional programming language, with non-strict semantics and strong static typing (Wikipedia contributors 2018).