Programs as Data
HelloLex

Niels Hallenberg

Thursday 2024-08-31

Mini lexer specification

We have a lexer generator specification: `hello.fsl`.

```
module Hello_fslex
open FSharp.Text.Lexing
open System
}

rule Tokenize = parse
  | ['0'-'9']      { LexBuffer<char>.LexemeString lexbuf }
  | _              { failwith "Lexer error: illegal symbol" }

{ // ending
[<EntryPoint>]
let main argv =
      printfn "Hello World from FsLex!\n\nPlease pass a digit:"
      let input = Console.ReadLine()
      let res=Tokenize (LexBuffer<char>.FromString input)
      printfn "The lexer recognizes %s" res
      0

}
```

Question 1

Read the specification `hello.fsl`.

What are the regular expressions involved, and which semantic values are they associated with?

Question 2

- Generate the lexer out of the specification using a command prompt. Which additional file is generated during the process?
- How many states are there by the automaton of the lexer?

  Hint: We assume you have followed one of the recommened setups such that everything can be build from the commandline. E.g.,

```
fslex --unicode hello.fsl
```

  You can get the number of the states of the automaton by reading the output when the lexer is generated.

Question 3

- Compile and run the generated program `hello.fs` from question 2.
  Hint: We assume you have followed one of the recommened setups such that everything can be build from the commandline. Below assumes executing the F# compiler directly and mono on mac.

```
% fsharpc -r ~/fsharp/FsLexYacc.Runtime.dll --standalone hello.fs
Microsoft (R) F# Compiler version 11.0.0.0 for F# 5.0
Copyright (c) Microsoft Corporation. All Rights Reserved.

% mono hello.exe
Hello World from FsLex!

Please pass a digit:
34
The lexer recognizes 3
%
```

Files in directory:

hello.fs

hello.fsl

Question 3

- Compile and run the generated program `hello.fs` from question 2.
  Hint: Now we assume you use `dotnet` command line tool.

Files in directory:
- hello.fs
- hello.fsl
- hello.fsproj

```
% dotnet build hello.fsproj
MSBuild version 17.9.4+90725d08d for .NET
  Determining projects to restore...
  Restored /Users/…/hello.fsproj (in 91 ms).
  hello -> /Users/…/bin/Debug/net8.0/hello.dll
Build succeeded.
    0 Warning(s)
    0 Error(s)
Time Elapsed 00:00:00.90
% dotnet bin/Debug/net8.0/hello.dll
Hello World from FsLex!

Please pass a digit:
34
The lexer recognizes 3
nh@MBP-Niels HelloLex %
```

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
    <Compile Include="hello.fs" />
  </ItemGroup>
  <ItemGroup>
    <Reference Include=
   "/Users/nh/fsharp/FsLexYacc.Runtime.dll" />
  </ItemGroup>
</Project>
```

Question 4

- Extend the lexer specification `hello.fsl` to recognize numbers of more than one digit. New lexer specification is `hello2.fsl`. Generate `hello2.fs`, compile and run the generated program.

```
% fslex --unicode hello2.fsl
compiling to dfas (can take a while...)
4 states
writing output
% dotnet build hello2.fsproj
MSBuild version 17.9.4+90725d08d for .NET
  Determining projects to restore...
  Restored /Users/…/hello2.fsproj (in 86 ms).
  hello2 -> /Users/…/bin/Debug/net8.0/hello2.dll
Build succeeded.
    0 Warning(s)
    0 Error(s)
Time Elapsed 00:00:00.90

% dotnet bin/Debug/net8.0/hello2.dll
Hello World from FsLex!
Please pass a digit:
234
The lexer recognizes 234
%
```

Question 5

- Extend the lexer specification `hello2.fsl` to recognize floating numbers. New lexer specification is `hello3.fsl`. Generate `hello3.fs`, compile and run the generated program.
  Hint: You can use the regular expression
  [+-]?([0-9]*[.])?[0-9]+
  to recognize floats.

```
% fslex --unicode hello3.fsl
compiling to dfas (can take a while...)
11 states
writing output
% dotnet build hello3.fsproj
MSBuild version 17.9.4+90725d08d for .NET
  Determining projects to restore...
  Restored /Users/…/hello3.fsproj (in 90 ms).
  hello3 -> /Users/…/bin/Debug/net8.0/hello3.dll
Build succeeded.
    0 Warning(s)
    0 Error(s)
Time Elapsed 00:00:00.90
% dotnet bin/Debug/net8.0/hello3.dll
Hello World from FsLex!

Please pass a digit:
23234.2323
The lexer recognizes 23234.232
```

Question 6

- Consider the 3 examples of input provided at the prompt and the result.
  Explain why the results are expected behaviour from the lexer.

```
% dotnet bin/Debug/net8.0/hello3.dll
Hello World from FsLex!

Please pass a digit:
34
The lexer recognizes 34

% dotnet bin/Debug/net8.0/hello3.dll
Hello World from FsLex!

Please pass a digit:
34.24
The lexer recognizes 34.24

% dotnet bin/Debug/net8.0/hello3.dll
Hello World from FsLex!

Please pass a digit:
34,34
The lexer recognizes 34
%
```