

Semester Project – Part 1 (aka Project 1)

First draft due for class on Wednesday, 20 September 2023
 Submit image(s) of your DFA as submissionidP1.jpg or submissionidP1.png
 If you have multiple images, use a, b, c, etc after the 1.

Preliminary Design Assignment

Design a DFA (*Deterministic Finite Automata*) that will recognize and categorize the following list of strings as defined by their regular expressions. α represents any alphabetical character # represents any digit.

Regular Expression	Category
+	PLUS_T
-	MINUS_T
/	DIV_T
*	MULT_T
=	EQUALTO_T
>=	GTE_T
<=	LTE_T
>	GT_T
<	LT_T
'	QUOTE_T

Regular Expression	Category
(LPAREN_T
)	RPAREN_T
(cad*r) (cd+r) (cd*ar)	LISTOP_T
(_ α)(α# _)*(? λ)	IDKEY_T
(+ - λ) #+	NUMLIT_T
(+ - λ) #*. #+	NUMLIT_T
(+ - λ) #+.#*	NUMLIT_T
(+ - λ) #+/#+	NUMLIT_T
". . ."	STRLIT_T

The full regular expression for the lexemes in our languages is:

`+ | - | / | * | = | >= | <= | > | < | ' | (|) | ((cad*r) | (cd+r) | (cd*ar)) | ((_|α)(α#|_)*(?|λ)) | ((+|-|λ) #+) | ((+|-|λ) #*. #+) | ((+|-|λ) #+.#*) | ((+|-|λ) #+/#+) | ". . ."`

A function that implements the algorithm would might read an input file and write a list of the strings found in the file and their categories. For example, if the input is:

```
car +cadr _abc-cons / if*else =and >= => "Hi all" abccadr
a_symbol 1 .123 -12.23 caddr? cdddr list? 124. "cadr" )( 1/2
```

The output might be:

```
LISTOP_T   car
PLUS_T     +
LISTOP_T   cadr
IDKEY_T    _abc
MINUS_T    -
```

```
IDKEY_T      cons
DIV_T        /
IDKEY_T      if
MULT_T       *
IDKEY_T      else
EQUALTO_T    =
IDKEY_T      and
GTE_T        >=
EQUALTO_T    =
GT_T         >
STRLIT_T     "Hi all"
IDKEY_T      Abccadr
IDKEY_T      a_symbol
NUMLIT_T     1
NUMLIT_T     .123
NUMLIT_T     -12.23
LISTOP_T     caddr
ERROR_T      ?
LISTOP_T     cddddr
IDKEY_T      list?
NUMLIT_T     124.
STRLIT_T     "cadr"
RPAREN_T     )
LPAREN_T     (
NUMLIT_T     1/2
```