

HW 5: Recursive Descent; LL Parsing

CS 440: Programming Languages and Translators

Due Sat Mar 30, 11:59 pm

No late assignments; solution will be posted Sun Mar 31 (Exam 2 is on Wed Apr 3)

Programming Problem [25 points]

1. Your job is to write a recursive descent parser for the grammar below. Each nonterminal should have a function that (tries to) parse it. Here's the language syntax (you may need to rearrange / refactor this)

$Expression \rightarrow Variable \mid Identifier \mid FunctionCall$

$Variable \rightarrow$ Upper case letter then zero or more alphanumeric or underscore

$Identifier \rightarrow$ Lower case letter then zero or more alphanumeric or underscore

$FunctionCall \rightarrow Identifier (Arguments)$

$Arguments \rightarrow Expression ArgTail \mid \epsilon$

$ArgTail \rightarrow , Expression ArgTail \mid \epsilon$

The file `HW_05_recdesc_skel.hs` contains the skeleton of a recursive descent parser. It's based on the implementation from Lecture 11, but there are some slight differences. For example, the parse result and parser are now parameterized datatypes, so the parsers are of type `Parser [Ptree]`. Names of structures have also been fiddled with. As in the lecture 11 parser, each nonterminal's parser should take an input string and yields either `FAIL` or `OK [parse tree] remaining_string`.

```
data Ptree = VAR String | ID String | FCN String [Ptree]
    deriving (Show, Eq, Read)
data Presult a = FAIL | OK a String deriving (Show, Eq, Read)
type Parser a = String -> Presult a
```

You're required to complete the skeleton implementation — you are not allowed use a parser-generator.

As usual, be sure to include your name(s) in the program you submit. A small warning: A submission that isn't syntactically accepted by the GHCi Haskell interpreter earns zero points.

Written Problem [25 points]

2. [25 = 9 + 8 + 8 points] Take the grammar below.

Rule 1: $S \rightarrow A \$$

Rules 2 - 4: $A \rightarrow a A \mid B b \mid \epsilon$

Rules 5 - 6: $B \rightarrow b B \mid c A$

- (a) Calculate the First, Follow, and $\rightarrow^* \epsilon$ sets / properties for the nonterminals.
- (b) Write out the LL(1) prediction parse table implied by part (a).
- (c) Use your parse table to complete the a trace of a parse of `a b c a c b b $`.

LL Parsing Trace

Stack (top is to left)	Input	Action
S		Begin parse
$A \$$	$a b c a c b b \$$	Rule 1: $S \rightarrow A \$$
$a A \$$	$a b c a c b b \$$	Rule 2: $A \rightarrow a A$
$A \$$	$b c a c b b \$$	Match a
	<i>... you complete ...</i>	