Wednesday, September 1, 2021 11:51 AM

top level / start symbol

$E \rightarrow T + E$

$E \rightarrow T$

$T \rightarrow F * T$

$T \rightarrow F$

$F \rightarrow NUM \mid ID \mid (E)$

A rule has

A LHS which is a non-terminal

A RHS (right hand side) which is a sequence of terminals and non-terminals

terminals = tokens

Terminals cannot appear on the LHS of a rule

Recursive descent parsing

- One parse function for each non-terminal
- when parse-X() is called:
  - Either a part of the input that corresponds to X is consumed, or
    no more no less
  - syntax_error is raised and we exit the program

Non-terminals X is consumed by calling parse-X()

Terminal tt is consumed by calling expect(tt);

In general

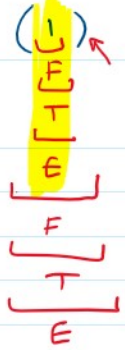
```
expect (ttype) :  t = gettoken();
                  if (t.token_type != ttype)
                      syntax_error();
                  else
                      return t;
```

```
parse_input()
{
    parse_E();
    expect (EOF);
}
```

1 + 2 * 3 ↑

1 + 2 * 3 ↑

```
parse_E()
{   // E → T + E
    // E → T
    parse_T();                    // consume T
    t = peek(1);
    if (t.token_type == PLUS)
    {
        expect (PLUS);            // consume +
        parse_E();                // consume E
    }
    else
    if ((t.token_type == EOF) ||
        (t.token_type == RPAREN))
        return;
    else
        syntax_error();
}
```




```
parse_F()
{   // F → NUM
    // F → ID
```

```
{   // F -> NUM
    // F -> ID
    // F -> ( E )

    t = peek(1);
    if (t.token_type == NUM)
        expect(NUM);                    // NUM
    else if (t.token_type == ID)
        expect(ID);                     // ID
    else if (t.token_type == LPAREN)
    {   expect(LPAREN);                 //  (
        parse_E();                      //  E
        expect(RPAREN);                 //  )
    } else
        syntax_error();
}

parse_T()
{ //T -> F * T
  //T -> F
    parse_F();
    t = peek(1);
    if (t.token_type == MULT)
    {   expect(MULT);
        parse_T();
    } else if((t.token_type == EOF) |
              (t.token_type == RPAREN) |
              (t.token_type == PLUS) )
        return;
    else
        syntax_error();
}
```