

## **LL(1) parser: top down parser.**

The first L stands for scan input Left to right.

The second L stands for constructs a Leftmost derivation

The 1 indicates that it uses one token of look-ahead.

- \* input buffer (hold the token stream)
- \* a single stack to store terminals and nonterminals yet to be parsed.
- \* a parsing table,  $PT[A, a]$ .
  - A is the set of non-terminals,
  - a is the set of terminals including \$ (for eof)

Abbreviations:

TOS: top of stack

S: Start symbol

T: set of terminals

NT: set of non terminals

input: current token

```
//initialize the stack:
```

```
push $;
```

```
push S;
```

```
while (not finished) {
```

```
    If (TOS is NT) {
```

```
        do PT [ ] lookup; // PT[TOS, input]
```

```
        If (there is an entry (a production rule)) {
```

```
            pop the NT from stack
```

```
            push the rule's right hand side to stack;
```

```
        }
```

```
        else report error; // bad syntax
```

```
    else if (TOS is T) {
```

```
        if (TOS matches input) {
```

```
            pop the T from stack;
```

```
            advance input;
```

```
        }
```

```
        else report error; // bad syntax
```

```
    else if (TOS is $) {
```

```
        if (input is $) report success;
```

```
        else report error;
```

```
    }
```