

Constructing a Parse Tree

Initial Conditions

Before we begin we need to have two things:

1. A Grammar
2. A string we wish to parse

So then what is the procedure:

1. Use the grammar to produce a derivation resulting in the given string
2. Use the derivation to produce the parse tree

Example 1:

We have the following grammar:

```
<S> ::= <round> <square> | <outer>
<round> ::= ( <round> ) | ( )
<square> ::= [ <square> ] | [ ]
<outer> ::= ( <outer> ] | ( <inner> ]
<inner> ::= ) <inner> [ | ) [
```

We will derive the following string: `(([])[[]]`

Example 1: Producing the derivation

A derivation is produce using the following steps:

1. Start with the start rule and select one of its options
2. We then continue to replace each rule until we reach a terminal working from left to right.
3. We repeat step 2 until all non-terminals are replaced by terminals, and we have produced the target string.

Goal: Derive (()) [[]]

```
<S> => <round> <square>
      => ( <round> ) <square>
      => ( ( ) ) <square>
      => ( ( ) ) [ <square> ]
      => ( ( ) ) [ [ ] ] --> Finished
```

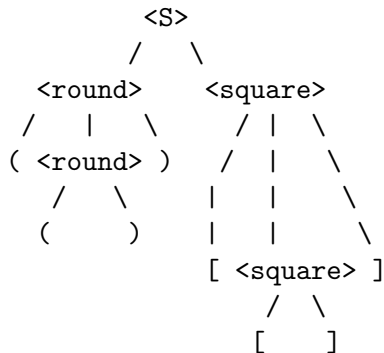
Example 1: Producing the parse tree

$\langle S \rangle \Rightarrow \langle \text{round} \rangle \langle \text{square} \rangle$

$$\Rightarrow (\langle \text{round} \rangle) \langle \text{square} \rangle$$

=> (()) <square>

=> (()) [<square>]

$$\Rightarrow (()) [[]]$$


Example 2

Grammar:

$$\langle S1 \rangle ::= \langle S1 \rangle + \langle S2 \rangle \mid \langle S2 \rangle$$
$$\langle S2 \rangle ::= \langle S2 \rangle * \langle S3 \rangle \mid \langle S3 \rangle$$
$$\langle S3 \rangle ::= (\langle S1 \rangle) \mid a \mid b \mid c$$

String: $a + b * c$

Example 2: Derivation

Goal: Derive $a + b * c$

$\langle S1 \rangle \Rightarrow \langle S1 \rangle + \langle S2 \rangle$
 $\Rightarrow \langle S2 \rangle + \langle S2 \rangle$
 $\Rightarrow \langle S3 \rangle + \langle S2 \rangle$
 $\Rightarrow a + \langle S2 \rangle$
 $\Rightarrow a + \langle S2 \rangle * \langle S3 \rangle$
 $\Rightarrow a + \langle S3 \rangle * \langle S3 \rangle$
 $\Rightarrow a + b * \langle S3 \rangle$
 $\Rightarrow a + b * c$

Example 2: Parse tree

$\langle S1 \rangle \Rightarrow \langle S1 \rangle + \langle S2 \rangle$

$\Rightarrow \langle S2 \rangle + \langle S2 \rangle$

$\Rightarrow \langle S3 \rangle + \langle S2 \rangle$

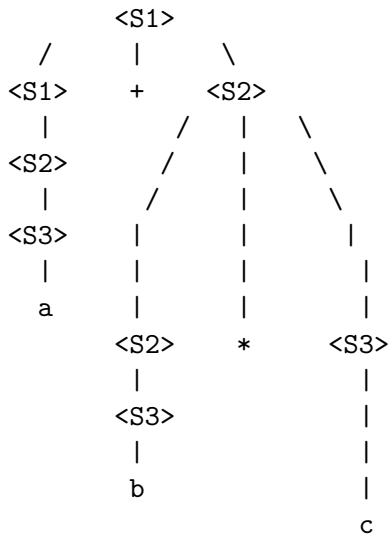
$\Rightarrow a + \langle S2 \rangle$

$\Rightarrow a + \langle S2 \rangle * \langle S3 \rangle$

$\Rightarrow a + \langle S3 \rangle * \langle S3 \rangle$

$\Rightarrow a + b * \langle S3 \rangle$

$\Rightarrow a + b * c$



We can also derive a string from the bottom up

Goal: Use Example 2 grammar to derive $a + b * c$

In This case we start with the rightmost terminal and continue to replace with non-terminals until we reach the start rule.

$\Rightarrow a + b * c$

$\Rightarrow a + b * \langle S3 \rangle$

$\Rightarrow a + \langle S3 \rangle * \langle S3 \rangle$

$\Rightarrow a + \langle S2 \rangle * \langle S3 \rangle$

$\Rightarrow a + \langle S2 \rangle$

$\Rightarrow \langle S3 \rangle + \langle S2 \rangle$

$\Rightarrow \langle S2 \rangle + \langle S2 \rangle$

$\Rightarrow \langle S1 \rangle + \langle S2 \rangle$

$\langle S1 \rangle \Rightarrow \langle S1 \rangle + \langle S2 \rangle$

We then build the parse tree starting from the bottom