

程序语言的语法

下面三组 C 程序语句中，每一组内的两句程序语句是相同的程序语句吗？第一组：

```
y=x+1; // 代码中的空格更少
```

```
y = x + 1; // 代码中的空格更多
```

第二组：

```
y = (x) + 1; // 代码中的包含多余的括号
```

```
y = x + 1; // 代码中无多余的括号
```

第三组：

```
y = 1 + x;
```

```
y = x + 1;
```

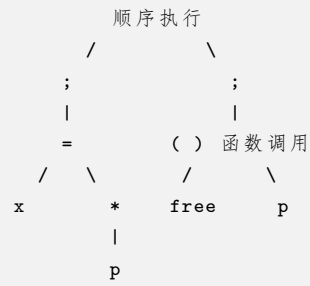
对于描述程序行为与程序正确性的理论而言，像第一组例子中的多余空格与第二组例子中的多余括号并不重要，因此，我们认为第一组与第二组都包含了相同的程序语句。而第三组中的两句程序语句则是不同的程序语句。

C 表达式 `* (p + 1) ++` 的结构：

```
++ (后缀运算符)
|
*
|
( )
|
+
/ \
p  1
```

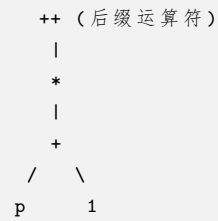
又例如，C 表达式 `x = * p; free(p);` 可以理解成为下面树结构。

C 表达式 `x = * p; free(p);` 的结构：



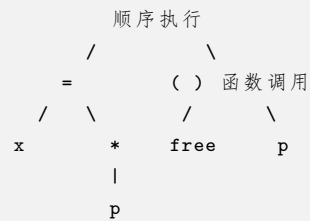
这些树结构中，有些信息是有些多余的。例如，上一个例子中的括号是多余信息。精简后可以得到以下树结构。

C 表达式 `* (p + 1) ++` 的结构：



类似的，C 表达式 `x = * p; free(p);` 的结构也可以简化。

C 表达式 `x = * p; free(p);` 的结构：



我们就把这样子精简之后的树结构称为这个程序的抽象语法树（Abstract Syntax Tree, AST）。