



数据结构与算法 (Python版)

树的应用：表达式解析 (上)

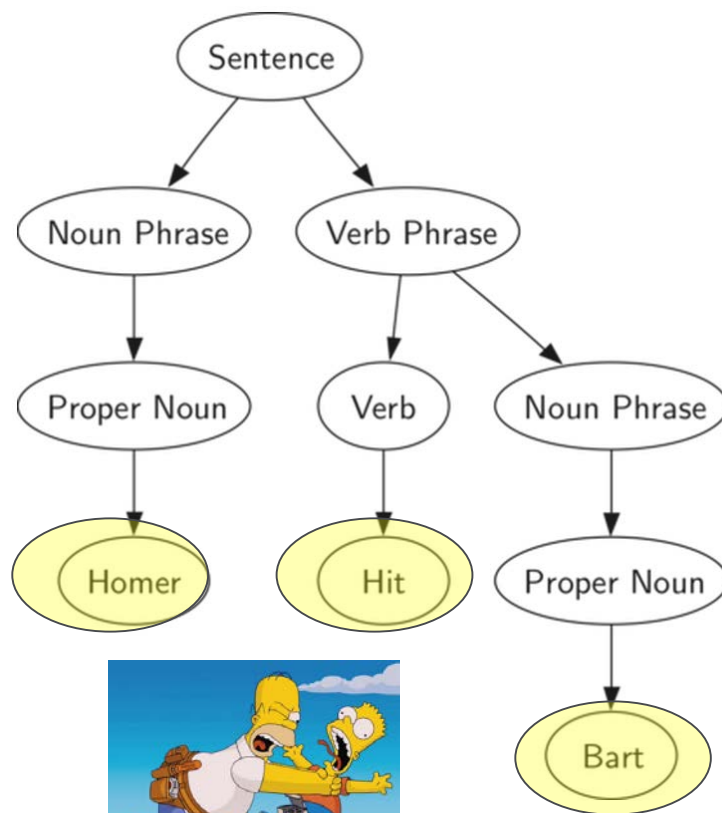
陈斌 北京大学 gischen@pku.edu.cn

树的应用：解析树（语法树）

- ❖ 将树用于表示语言中句子，可以分析句子的各种语法成分，对句子的各种成分进行处理

- ❖ 语法分析树

主谓宾，定状补



树的应用：解析树（语法树）

❖ 程序设计语言的编译

词法、语法检查

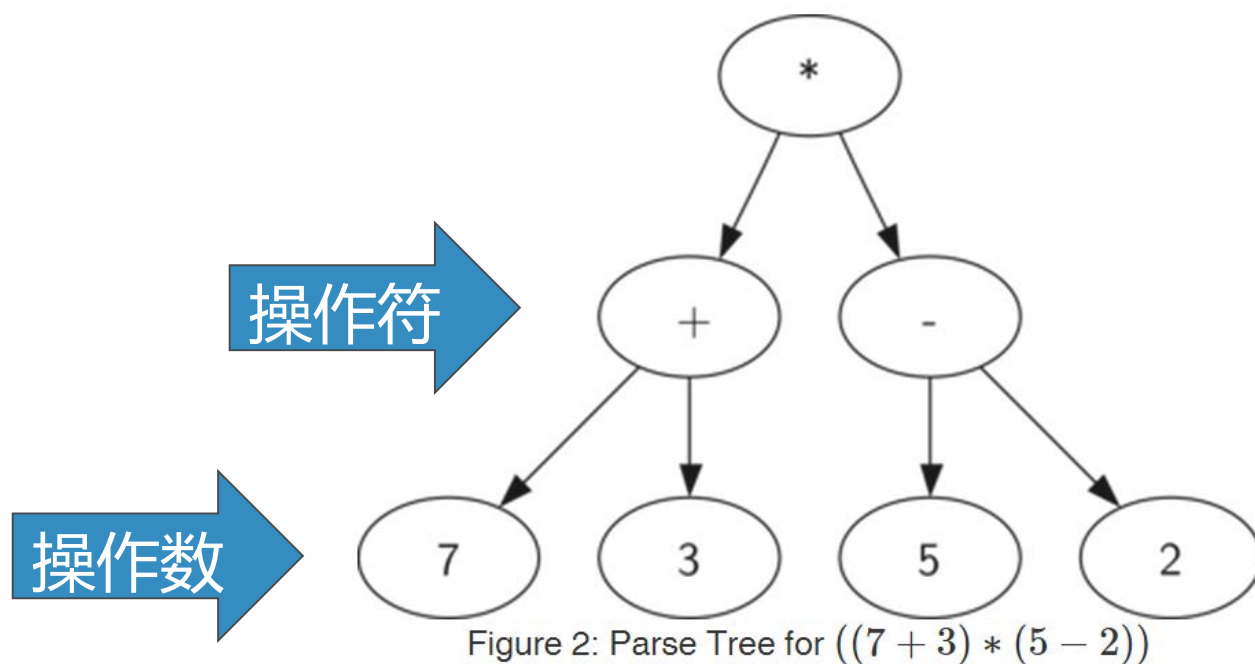
从语法树生成目标代码

❖ 自然语言处理

机器翻译、语义理解

树的应用：表达式解析

- ❖ 我们还可以将表达式表示为树结构
叶节点保存操作数，内部节点保存操作符



树的应用：表达式解析

❖ 全括号表达式 $((7+3)*(5-2))$

由于括号的存在，需要计算 $*$ 的话，就必须先计算 $7+3$ 和 $5-2$ ，表达式层次决定计算的优先级

越底层的表达式，优先级越高

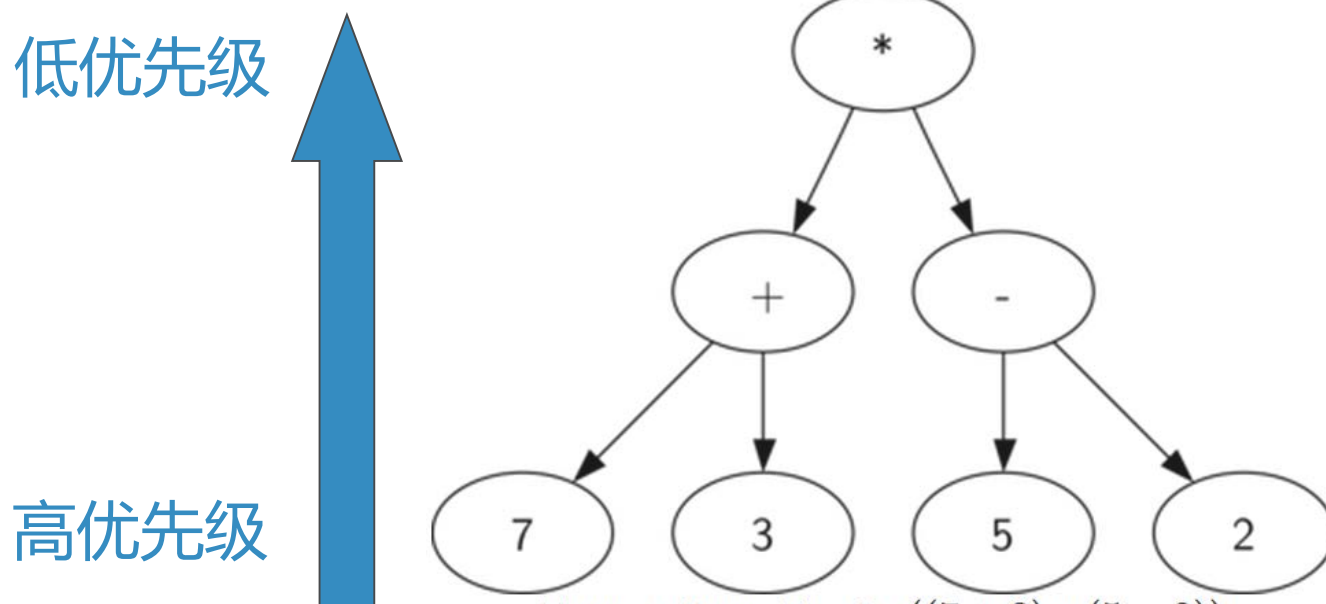
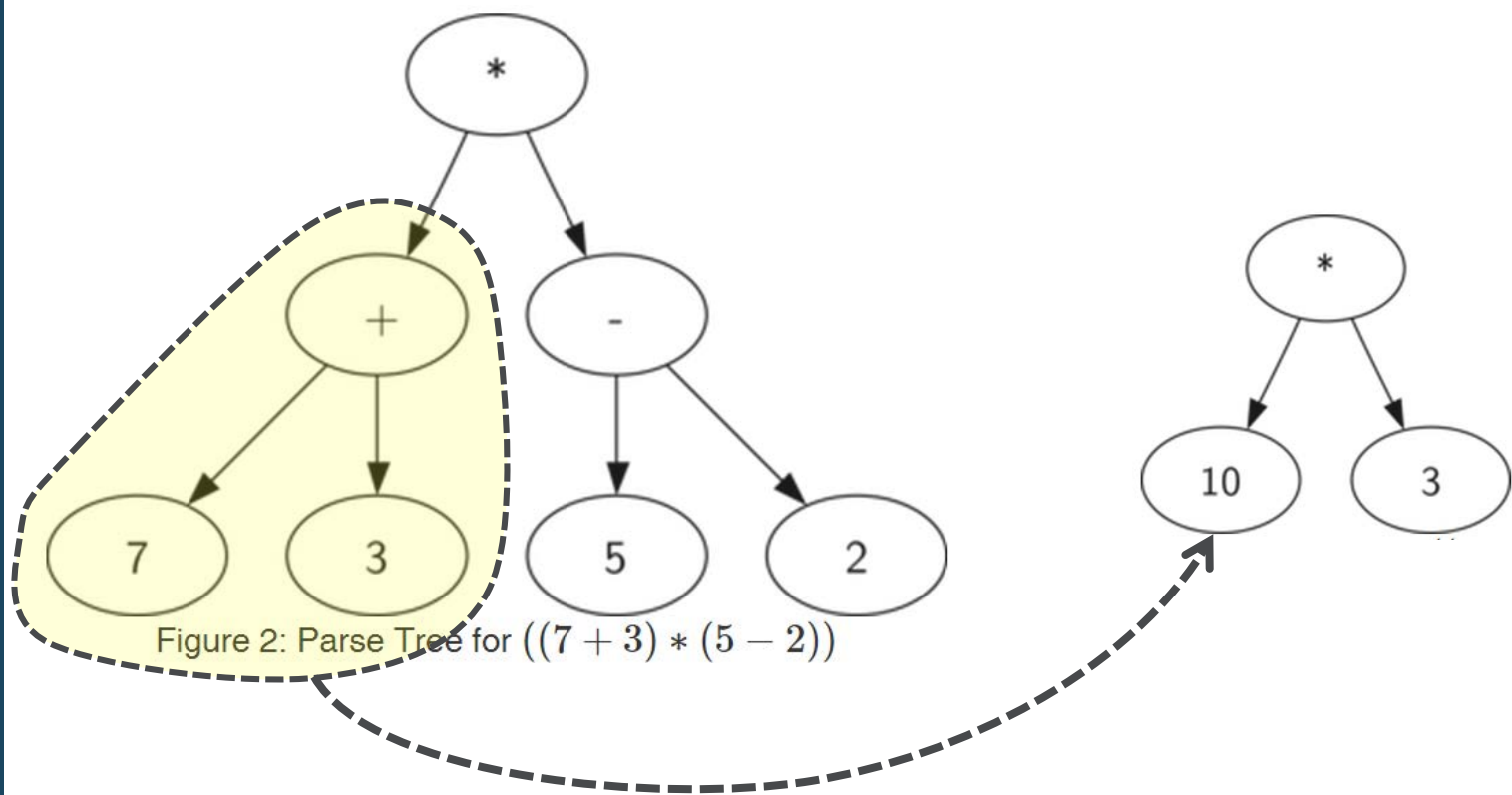


Figure 2: Parse Tree for $((7+3)*(5-2))$

树的应用：表达式解析

❖ 树中每个子树都表示一个子表达式

将子树替换为子表达式值的节点，即可实现求值



表达式解析树

❖ 下面，我们用树结构来做如下尝试

从全括号表达式构建表达式解析树

利用表达式解析树对表达式求值

从表达式解析树恢复原表达式的字符串形式

❖ 首先，全括号表达式要分解为**单词Token列表**

其单词分为括号“()”、操作符“+ - * /”和操作数“0~9”这几类

左括号就是表达式的开始，而右括号是表达式的结束

建立表达式解析树：实例

❖ 全括号表达式：(3+(4*5))

分解为单词表

['(', '3', '+', '(', '4', '*', '5',
')', ')']

(3+(4*5))

['(', '3', '+', '(', '4', '*', '5', ')', ')']

建立表达式解析树：实例

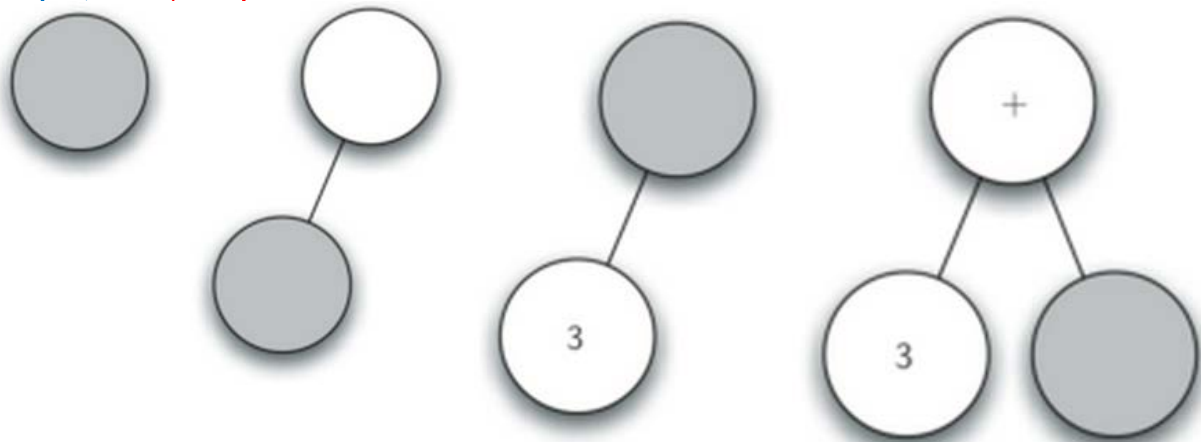
❖ 创建表达式解析树过程

创建空树，当前节点为根节点

读入 '(', 创建了左子节点，当前节点下降

读入 '3', 当前节点设置为3，上升到父节点

读入 '+', 当前节点设置为+, 创建右子节点，当前节点下降



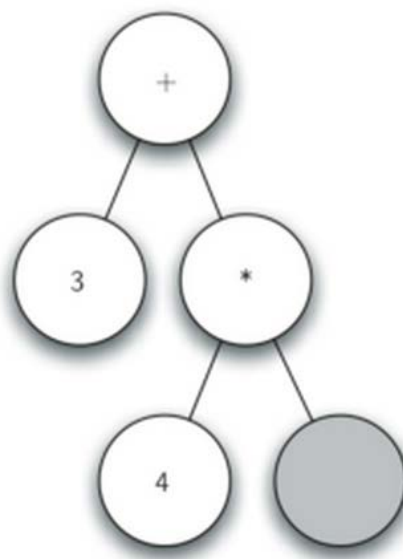
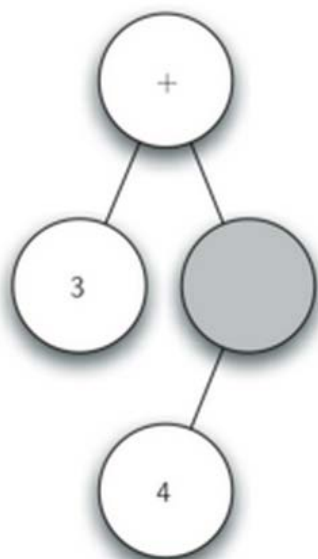
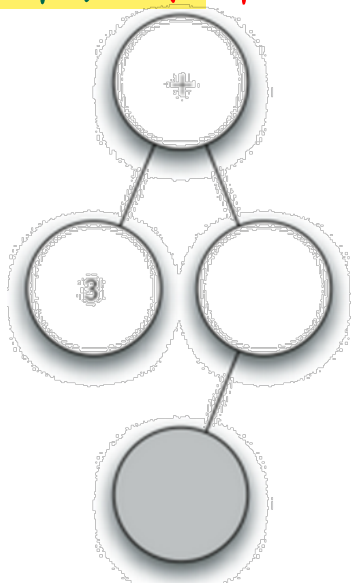
建立表达式解析树：实例

❖ 创建表达式解析树过程

读入 '(', 创建左子节点, 当前节点下降

读入 '4', 当前节点设置为4, 上升到父节点

读入 '*', 当前节点设置为*, 创建右子节点, 当前节点下降



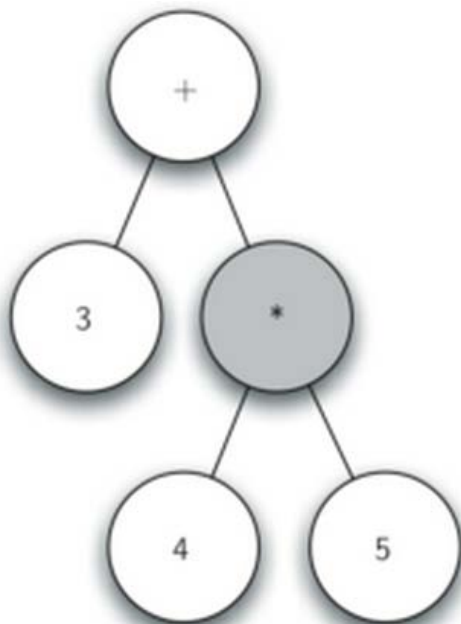
建立表达式解析树：实例

❖ 创建表达式解析树过程

读入 '5', 当前节点**设置**为5, **上升**到父节点

读入 ')', **上升**到父节点

读入 ')', 再**上升**到父节点



建立表达式解析树：规则

❖ 从左到右**扫描**全括号表达式的每个**单词**，依据规则建立解析树

如果当前单词是"**(**"：为当前节点添加一个新节点作为其左子节点，当前节点下降为这个新节点

如果当前单词是**操作符**"**+**，"**-**，"**/**，"*****"：将当前节点的值设为此符号，为当前节点添加一个新节点作为其右子节点，当前节点下降为这个新节点

如果当前单词是**操作数**：将当前节点的值设为此数，当前节点上升到父节点

如果当前单词是"**)**"：则当前节点上升到父节点

