

Applications of Tree Traversals

- example: creating a table of contents using preorder traversal
- redesign a top-down recursion that includes the current depth as additional parameter

```
def preorder_indent(T, p, d):
    // print preorder of subtree of T rooted at p at depth d
    print(2*d*' ' + str(p.element()))
    for c in T.children(p):
        preorder_indent(T, c, d+1)
```

Parenthetic Representation

- parenthetic string representation is defined by

$$P(T) = \text{str}(p.\text{element}())$$

// custom traversal to print parenthetic string

```
def parenthesize(T, p):
    print(p.element(), end='')
    // use of ends avoids trailing newline
    if not T.is_leaf(p):
        first_time = True
        for c in T.children(p):
            sep = '(' if first_time else ','
            // determine proper separator
            print(sep, end='')
            first_time = False
            parenthesize(T, c)
            // recur on child
        print(')', end='')
        // closing parenthesis
```

Computing Disk Space

- recursive computation of disk space is emblematic of a postorder traversal
- in this case, we need a mechanism for children to return info to parent

```
def disk_space(T, p):
    subtotal = p.element().space()
    // space at position p
    for c in T.children(p):
        subtotal += disk_space(T, c)
        // add child's space to subtotal
    return subtotal
```