

# The geeky thrill of discovering that two things are really the same thing, just with different labels

 [devblogs.microsoft.com/oldnewthing/20140414-01](http://devblogs.microsoft.com/oldnewthing/20140414-01)

April 14, 2014



Raymond Chen

Today's post about binomial coefficients was intended to be a warm-up for Catalan numbers, but it turns out Eric Lippert already covered them, first in the context of binary trees, then in the context of arbitrary trees and forests, and then again in the context of matched parentheses. Another way of seeing the correspondence between forests and matched parentheses is simply to consider each `{` as an XML open-tag and each `}` as an XML end-tag.

One thing to take away from the enumeration of objects controlled by Catalan numbers is that when you see multiplication in a recurrence relation, that typically corresponds to a nested loop. (We saw this ourselves when we studied Stirling numbers of the second kind.)

The correspondence between binary trees and arbitrary forests is done by simply renaming variables: `leftChild` and `rightChild` turn into `firstChild` and `nextSibling`.

Renaming variables also reveals an interesting equivalence between the two algorithms for reversing a linked list. One technique is to do link rewriting:

```
Node *Reverse(Node *head)
{
    Node *prev = nullptr;
    while (head) {
        // The node we are rewriting
        Node *current = head;
        // Advance to next node before
        // we overwrite the outbound pointer
        head = current->next;
        // Reprint to previous node
        current->next = prev;
        // Advance the trailing pointer
        prev = current;
    }
    return prev;
}
```

Another technique is to pop nodes off one list while pushing them onto another.

```
Node *Reverse(Node *head)
{
    Node *result = nullptr;
    while (head) {
        // Pop
        Node *current = head;
        head = current->next;
        // Push
        current->next = result;
        result = current;
    }
    return result;
}
```

But if you look more closely at the two versions, you'll see that they are not really two algorithms. They are the *same* algorithm, just with different comments and variable names!

One of my colleagues used this as an interview question and guided candidates through both algorithms, only to discover later that they were actually the same algorithm, merely viewed through different-colored glasses.

[Raymond Chen](#)

**Follow**

