Consecutive Right-Angle Brackets

In the context of template argument lists, `>>` is parsed as two separate closing angle brackets.

Description

Prior to C++11, a pair of consecutive right-pointing angle brackets anywhere in the source code was always interpreted as a bitwise right-shift operator, so making an intervening space was required if the brackets were to be treated as separate closing-angle-bracket tokens:

```
// C++03
std::vector<std::vector<int>> v0;   // annoying compile-time error in C++03
std::vector<std::vector<int> > v1;  // OK
```

To facilitate the common use case above, a special rule was added whereby, when parsing a template-argument expression, *non-nested* — i.e., not placed within parentheses — appearances of `>`, `>>`, `>>>`, and so on are to be treated as ~~separate closing angle brackets~~:

```
// C++11
std::vector<std::vector<int>> v0;               // OK
~~std::vector<std::vector<std::vector<int>>> v1;  // OK~~
```

Using the greater-than or right-shift operators within template-argument expressions

For templates that take only type parameters, there’s no issue. When the template parameter is a non-type, however, the greater-than or right-shift operators might be useful. In the unlikely event that we need either the greater-than operator, `>`, or the right-shift operator, `>>`, within a non-type template-argument expression, we can achieve our goal by nesting that expression within parentheses:

```
const int i = 1, j = 2;  // arbitrary integer values (used below)

template <int I> class C { /*...*/ };
    // class C taking non-type template parameter I of type int

C<i > j>    a1;  // Error, always has been
C<i >> j>   b1;  // Error in C++11, OK in C++03
C<(i > j)>  a2;  // OK
C<(i >> j)> b2;  // OK
```

In the definition of `a1` above, the first `>` is interpreted as a closing angle bracket, and the subsequent `j` is and always has been a syntax error. In the case of `b1`, the `>>` is, as of C++11, parsed as a pair of separate tokens in this context, so the second `>` is now considered an error.