

Want Speed?
Pass by Value to StringAlgo

The Idea

Turn this:

```
Sequence to_upper_copy(const Sequence& Input) {  
    return Sequence(  
        make_transform_iterator(begin(Input), to_upperF()),  
        make_transform_iterator(end(Input), to_upperF())  
    );  
}
```

The Idea

... into this:

```
Sequence to_upper_copy(Sequence Input) {  
    to_upper(Input); // in-place  
    return Input;  
}
```

The Idea

- Argument passing from rvalue
 - Copy is elided completely
 - Or else the argument is moved
- Argument passing from lvalue
 - Copy would have been done anyway

The Test

- Construct large string or many small strings
- Measure several nested calls to StringAlgo functions

The Results

- For large string, no difference was found
- For many small strings, modified version was a little faster
- No difference on VS 11, Win32
- From 1.3 to 1.1 seconds on VS 11, Win64
- From 1.7 to 1.0 seconds on Clang/libc++, MacOS X 64-bit

The Results

- But: passing from lvalue not measured!
- Might have been pessimized

Compare

```
Sequence to_upper_copy(const Sequence& Input) {  
    return Sequence(  
        make_transform_iterator(begin(Input), to_upperF()),  
        make_transform_iterator(end(Input), to_upperF())  
    );  
}
```

```
Sequence to_upper_copy(Sequence Input) {  
    to_upper(Input); // in-place  
    return Input;  
}
```