Subject: String improvements
Posted by hojtsy on Sat, 25 Feb 2006 17:12:23 GMT
View Forum Message <> Reply to Message

I added a feature to String::Find and String::ReverseFind that position could be also given relative to the end of the String, with -1 being equal to the length.

s.Find('R', -3) == s.Find('R', s.GetCount()+1-3);
s.ReverseFind('R', -3) == s.ReverseFind('R', s.GetCount()+1-3);

Implementation is below:template <class T, class S>
int AString<T, S>::Find(int chr, int from) const
{
 ASSERT(from >= -GetLength()-1 && from <= GetLength());
 if(from < 0)
  from += GetLength()+1;
 const T *e = End();
 for(const T *s = ptr + from; s < e; s++)
  if(*s == chr)
   return s - ptr;
 return -1;
}

template <class T, class S>
int AString<T, S>::ReverseFind(int chr, int from) const
{
 ASSERT(from >= -GetLength()-1 && from <= GetLength());
 if(from < 0)
  from += GetLength()+1;
 const T *s = ptr + from;
 while(--s >= ptr)
  if(*s == chr)
   return s - ptr;
 return -1;
}
This makes the AString::ReverseFind(int chr) unnecessary, because the ReverseFind(int chr, int from) could have -1 as default value for "from". Could you please add this improvement to uppsrc?

There is one trick, but it comes from the old implementation. If you invoke Find('R', x) it will find 'R' if it present at pos x, but ReverseFind('R', x) only starts to check at position x-1. So to start reverese checking from the last char you can use ReverseFind('R', -1), but to start forward checking from the last char you need to use Find('R', -2). Even though I find this unintuitive this is the existing behaviour of Find and ReverseFind for positive numbers too and changing it would break existing implementations. What is your opinion about changing this behaviour so that ReverseFind('R', x) starts checking at position x, and not x-1?

By the way, why is the chr parameter int? Shouldn't it rather be T?