Engineering Informatics 1
Recursion

Dr. Jonas Pföh

WS 2013-2014
16.01.2014
Recursion is the process of solving a problem by dividing the problem and solving (multiple) smaller sub-problem(s) of the same kind. In practice we often see these types of problems solved by functions that divide a problem into smaller sub-problem(s), then call themselves to solve these sub-problem(s).
When we talk about recursion, we must define a base case. This is the smallest possible sub-problem to be solved and often has a simple solution, which will be returned to help reconstruct the solution of the larger problem.
The *factorial* of a positive integer is the product of all positive integers less than or equal to that integer.

\[ 3! = 3 \times 2 \times 1 = 6 \]
An Example - Factorial (cont)

How might we define this as a function $f$?
How might we define this as a function $f$?

$$f(x) = \prod_{i=1}^{x} i$$
An Example - Factorial (cont)

How might we define this as a function $f$?

$$f(x) = \prod_{i=1}^{x} i$$

This would be an example of an iterative solution and is completely valid, however can we solve this by dividing it into smaller sub-problems?
What about this:

\[ f(x) = \begin{cases} 
1 & \text{if } x = 1, \\
x \times f(x - 1) & \text{if } x > 1 
\end{cases} \]
An Example - Factorial (cont)

```c
int factorial(int x) {
  if (x < 1) return -1;
  if (x == 1) return 1;
  else return x * factorial(x-1);
}
```
More Examples

- Fibonacci Numbers
- Palindrome Test