



Semantics, WS 2003 – Assignment 1

Prof. Dr. Gert Smolka, Dipl.-Inform. Guido Tack

<http://www.ps.uni-sb.de/courses/sem-ws03/>

Recommended reading: Types and Programming Languages, chapters 1–5, emphasis on 5

The exercises on this assignment sheet are all about Church numerals. In order to test your solutions, you should try them directly with your favourite ML dialect! We provide you with two conversion functions from and to ordinary integers, which should make it easy for you to generate tests:

```
fun scc n = fn s => fn z => s (n s z)
```

```
fun church 0 = (fn s => fn z => z)  
  | church n = scc (church (n-1))
```

```
fun int n = n (fn x => x + 1) 0
```

Exercise 1.1: Church numerals – Power Define a term for raising one number to the power of another.

Exercise 1.2: Church numerals – Subtraction Define a term for subtracting one number from another number. You can use `prd`.

Exercise 1.3: Church numerals – Equality Write a function `equal` that tests two numbers for equality and returns a church boolean.

Exercise 1.4: Church numerals – Factorial Write a combinator `fac` that computes factorials (i.e., yields $c_n!$ for c_n). You can use `pair`, `fst`, `snd`, `scc`, and `times`.

Exercise 1.5: Church numerals – Lists See exercise 5.2.8 in *Types and Programming Languages*.

Exercise 1.6: Church numerals – Sum of list Use `fix` and the encoding of lists from the previous exercise to write a function that sums lists of Church numerals.