Assignment 1
Semantics, WS 2007/08

Prof. Dr. Gert Smolka, Dr. Jan Schwinghammer
www.ps.uni-sb.de/courses/sem-ws07/

Standard ML: recursion, cascaded procedures

Exercise 1.1 (Fib from iter) Write a procedure $fib : int \to int$ such that

$$fib\ n = \text{if } n < 2 \text{ then } n \text{ else } fib(n-1) + fib(n-2)$$

Use the procedure

```
fun iter n x f = if n<1 then x else iter (n-1) (f x) f
```

to obtain the necessary recursion.

Exercise 1.2 (Gcd from loop) Consider the procedure

```
fun loop x p f = if p x then loop (f x) p f else x
```

Write a procedure $gcd : int \to int \to int$ that yields the greatest common divisor for positive arguments. Use the equation

$$gcd\ x\ y = if\ x = 0\ then\ y\ else\ gcd(y\ mod\ x)\ x$$

and use *loop* to realize the necessary recursion.

Exercise 1.3 (Mapr from foldl) Write a procedure $mapr : (\alpha \to \beta) \to \alpha\ list \to \beta\ list$ such that

$$mapr\ f\ [x_1, \ldots, x_n] = [f\ x_n, \ldots, f\ x_1]$$

To obtain the necessary recursion, use the following procedure:

```
fun foldl f s nil = s
  | foldl f s (x::xr) = foldl f (f(x,s)) xr
```

Exercise 1.4 (List from iter) Write a procedure $list : int \to (int \to \alpha) \to \alpha\ list$ such that

$$list\ n\ f = [f\ 1, \ldots f\ n]$$

using *iter* instead of a recursive definition.

Exercise 1.5 (Power from fix) Write a procedure $power : int \to int \to int$ such that $power\ x\ n = x^n$. Use *fix* to realize the necessary recursion, where

```
fun fix f x = f (fix f) x
```

Exercise 1.6 (Cas and car) Write procedures

$$cas : (\alpha * \beta \to \gamma) \to \alpha \to \beta \to \gamma$$
$$car : (\alpha \to \beta \to \gamma) \to \alpha * \beta \to \gamma$$

such that $car(cas\ f) = f$ and $cas(car\ g) = g$.

Exercise 1.7 (Cond) Write a procedure $cond : bool \to (unit \to \alpha) \to (unit \to \alpha) \to \alpha$ such that

$$cond\ t_0\ (fn\ () \Rightarrow t_1)\ (fn\ () \Rightarrow t_2)$$

is semantically equivalent to the expression $if\ t_0\ then\ t_1\ else\ t_2$.