



## Einführung in die Informatik 2

Prof. Dr. Andrey Rybalchenko, M.Sc. Ruslán Ledesma Garza

## Nützliche Hinweise zur Nachholklausur

$$\frac{T(b) = t}{T \vdash b : v} \quad \frac{k \in \mathbb{Z}}{T \vdash k : \text{int}} \quad \frac{T \vdash e_1 : t_1 \rightarrow t_2 \quad T \vdash e_2 : t_1}{T \vdash e_1 e_2 : t_2}$$

$$\frac{T \vdash e_1 : t_1 \quad T \vdash o : t_1 * t_2 \rightarrow t \quad T \vdash e_2 : t_2}{T \vdash e_1 o e_2 : t} \quad \frac{T \vdash e_1 : \text{bool} \quad T \vdash e_2 : t \quad T \vdash e_3 : t}{T \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : t}$$

$$\frac{T + [f := t_1 \rightarrow t_2] + [b := t_1] \vdash e : t_2}{T |> (\text{fun } f (b : t_1) : t_2 = e) : T + [f := t_1 \rightarrow t_2]} \quad \frac{T \vdash e : t}{T |> \text{val } b = e : T + [b := t]}$$

$$\frac{T_0 |> d_1 : T_1 \quad \dots \quad T_n |> d_n : T_{n+1}}{T_0 |> d_1 \dots d_n : T_{n+1}}$$

```
fun map f nil      = nil
  | map f (x::xr) = (f x) :: (map f xr)
```

```
map : ('a -> 'b) -> 'a list -> 'b list
```

```
fun filter f nil      = nil
  | filter f (x::xr) = if f x then x :: filter f xr
  else filter f xr
```

```
filter : ('a -> bool) -> 'a list -> 'a list
```

```
fun exists f nil      = false
  | exists f (x::xr) = f x orelse exists f xr
```

```
exists : ('a -> bool) -> 'a list -> bool
```

```
fun all f nil      = true
  | all f (x::xr) = f x andalso all f xr
```

```
all : ('a -> bool) -> 'a list -> bool
```

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

```
foldl : ('a * 'b -> 'b) -> 'b -> 'a list -> 'b
```

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

```
foldr : ('a * 'b -> 'b) -> 'b -> 'a list -> 'b
```

```
fun length nil      = 0
  | length (x::xr) = 1 + length xr

length : 'a list -> int

fun nth nil n = raise Subscript
  | nth _ ~1 = raise Subscript
  | nth (x :: xs) 0 = x
  | nth (x :: xs) n = nth xs (n - 1)

nth : 'a list -> int -> 'a
```