

Isabelle/HOL Exercises

Lists

Sum of List Elements, Tail-Recursively

- (a) Define a primitive recursive function *ListSum* that computes the sum of all elements of a list of natural numbers.

Prove the following equations. Note that $[0..n]$ und *replicate* *n* *a* are already defined in a theory *List.thy*.

```
consts ListSum :: "nat list  $\Rightarrow$  nat"
```

```
theorem "2 * ListSum [0.. $n+1$ ] = n * (n + 1)"
```

```
theorem "ListSum (replicate n a) = n * a"
```

- (b) Define an equivalent function *ListSumT* using a tail-recursive function *ListSumTAux*. Prove that *ListSum* and *ListSumT* are in fact equivalent.

```
consts ListSumTAux :: "nat list  $\Rightarrow$  nat  $\Rightarrow$  nat"
```

```
consts ListSumT :: "nat list  $\Rightarrow$  nat"
```

```
theorem "ListSum xs = ListSumT xs"
```