as parameters a reduction operator, an initial accumulator value, and a sequence of elements. The accumulator need not have the same type as the elements of the sequence. For instance, we can use a reduction to count the number of occurrences of a particular value in a sequence. $[S, T] = \underbrace{reduce : (S \times T \to S) \times S \times \operatorname{seq} T \to S}$ $\forall f: S \times T \rightarrow S \bullet \forall s: S \bullet \forall ts : \text{seq } T \bullet$ reduce(f, s, ts) =**if** # ts = 0then s

We introduce a generalised linear reduction over sequences. The function takes

then s else reduce(f, f(s, head ts), tail ts)An efficient implementation could perform a tree-like reduction on vector hardware.