Proof Techniques

Proofs on Infinite Lists

For proofs on infinite list the approx function is very helpful.

```
approx               :: Integer -> [a] -> [a]
approx (n+1) []      = []
approx (n+1) (x:xs) = x:approx n xs
```

- $\lim_{n\to\infty}$ `approx n xs` $=$ `xs`
- $\forall$`n`(`approx n xs` $=$ `approx n ys`) $\implies$ (`xs` $=$ `ys`). That is, induction can be used with `approx` in order to make assertions about infinite lists.

Fixpoint

To find the fixpoint of a function f we do $f^n(\perp)$ for increasingly larger n until the returned value repeats or starts cycling. In the first case the repeated value is the fixpoint, in the later we have that there is no fixpoint (the fact it cycles also serves as the non-existence proof).

Bottom $\perp$

Some useful tips about $\perp$:

- $\perp$ is always $\perp_?$ for some type ? which you determine from the context in which $\perp$ is used.
- $\perp_? \sqsubseteq \perp_{[?]} \sqsubseteq \perp_{[[?]]} \cdots$