Polymorphic inductive definitions can be thought of as what?

What are the types of the *polymorphic constructors* `nil` and `cons`?

With respect to polymorphic inductive types, Coq will automatically infer what?

What won't Coq automatically infer?

How can we avoid writing type arguments in an invocation?

Throughout a module?

At declaration site?

How can you use explicit type arguments after having requested inference?Use `@`.

What's the difference between `(x,y)` and `(X*Y)`?

Which way does the type arrow associate?

What are the type signatures of currying and uncurrying a function?

Give the syntax of anonymous functions.

Define a function that overrides other functions whose domain is `nat`s.

Give its type.

Describe the `unfold` and `fold` tactics.

What does it mean to say that constructors in Coq are *disjoint*?

What does it mean to say that constructors in Coq are *injective*?

Describe the `inversion` tactic.

How can tactics be applied to hypothesis instead of the goal?

Explain the difference between *forward* and *backward* reasoning.

What happens if you `destruct` on an expression instead of a value?

Explain the `remember` tactic.

Explain the `apply ... with ...` tactic.