What is backwards reasoning?

Why does Prolog use it?

How can you load a Prolog database for
interactive use?

What is the closed world assumption?

What is a structure?

What is a base clause?

What is a nonbase clause?

What is a predicate?

What is a program?

How can you include special characters and spaces in an atom, or begin it with a capital letter?

What are double quotes for?

How can you include escape characters in a quoted atom?

What are the four ports of a structure?

Prolog's logic is *non-monotonic*. What does that mean?

Why might you need to double the parens of functor?

What are the limitations of backtracking.

How do you write to stdout?

How can you view the available facts and rules?

How can you view the available structures for a predicate?

What does a single underscore (_) do?

What's the difference between cut (`!`) and `fail`?

What happens if you combine cut and fail?

Why might you start a variable with an underscore?

What is the scope of a variable?

What order are clauses of a predicate tried in?

What can you use in place of `if`?

What can you use in place of loops?

How can you get around the lack of functions
with no return values?

How can you get around the lack of functions
with return values?

How can you work around the lack of
assignable state?

What is the $=..$ operator?

What predicate describes list membership?

What predicate describes combining lists?

How can you force a predicate to succeed?

What should you do if you don't care if a call
succeeds and don't want failure to cause
backtracking?

What are `asserta` and `assertz` for?

What does `abolish` do?

What's the catch with arithmetic in Prolog?

What is the difference between static and
dynamic clauses in SWI Prolog?