What is backwards reasoning?

Why does Prolog use it?

Backwards reasoning is from (potential) conclusions to facts instead of from facts to conclusions.

Prolog uses it because the space of possible conclusions grows too quickly in the number of premises.

How can you load a Prolog database for interactive use?

You can use the `consult` and `reconsult` predicates, or use bracket notation.

Bracket notation allows re-consultation of several files at once:

```
['file1.pl', file2.pl']
```

What is the closed world assumption?

A Prolog database knows everything it needs to know.

What is a structure?

A name followed by zero or more arguments. Parens are omitted if there are no arguments.

What is a base clause?

A structure terminated by a period. It represents a simple fact.

What is a nonbase clause?

A structure followed by a turnstile and a list of structures separated by commas. It represents a rule.

What is a predicate?

A **collection** of clauses with the same *functor* (name) and arity.
This is similar to the collection of overloaded versions of a function in an imperative language.

What is a program?

A collection of predicates, in any order.

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

Use single quotes, which does **not** make a string.

How can you include escape characters in a quoted atom?

Use double quotes or an escaped single quote to use a single quote.
Other escapes use backslash as in other languages.

What are the four ports of a structure?

```
         -----
call --> |   | --> exit
fail <-- |   | <-- redo
         -----
```

`exit` ports connect to `call` ports.
`fail` ports connect to `redo` ports.

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

Facts can be added at any time using the `assert` predicate.
Facts can be removed at any time using the `retract` predicate.
Such rules are *dynamic*.

Why might you need to double the parens of functor?

To force a rule to be interpreted as a single argument, since rules contain commas.

```
assert((loves(chuck, X) :- female(X), rich(X))).
```

What are the limitations of backtracking.

- Output can't be undone.
- `assert` and `retract` can't be undone either.

How do you write to stdout?

`write` predicate outputs its single argument to stdout.
`nl` writes a newline.

How can you view the available facts and rules?

How can you view the available structures for a predicate?

```
listing(predicate)

listing
```

What does a single underscore (_) do?

It's an anonymous variable and can represent any term.

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

`fail` doesn't force other the entire predicate to fail. Other clauses will be tried.

Using a cut creates a commit point, preventing backtracking past the commit point and preventing attempts on other clauses.

What happens if you combine cut and fail?

The predicate as a whole fails.