Implement a LISP function, `derivable(Premises, Prop)` where `Premises` denotes a list of formulas in Propositional Logic and `Prop` denotes a proposition, to determine whether `Prop` is derivable from `Premises` or not. First you need to construct an AND/OR graph from `Premises` and then search that graph for a solution.

Assume that the list of proposition names is given. A formula in `Premises` can be of two forms:

- a proposition

- $A \rightarrow P$ where $P$ is a proposition and $A$ is a logical formula that involves logical AND, OR with parentheses. As far as the representation is concerned, $A \rightarrow P$ is given as a list that contains three elements, ($A$ implies $P$). Here $A$ can be a proposition or a list of three elements where the first and third elements can be a formula similar to $A$ and the second element can be and or or. Some examples:

  \[
  p \rightarrow q: (p \implies q) \\
  p \lor q \rightarrow r: ((p \lor q) \implies r) \\
  (p \land q) \lor r \rightarrow t: (((p \land q) \lor r) \implies t)
  \]

The function will return the solution AND/OR subtree (in list structure) if the `Prop` derivable from the `Premises`, nil otherwise.

An Example Call:

```lisp
> derivable('((p and q) implies t)
  ((p and v) or t) implies s))
\"s\")
```