Exercise: implementing temporal properties

1. Returns false until A occurs, then returns true from the subsequent instant onward

   ```
   node after(a : bool) returns (o : bool);
   -- make clean; make MAIN=after TRACE=trace1.txt
   ```

2. Returns true if and only if its first input has been continuously true since the last time its second input was true

   ```
   node always_since(b, a : bool) returns (o : bool);
   -- make clean; make MAIN=always_since TRACE=trace2.txt
   ```

3. Returns true if and only if its first input has been true at least once since the last time its second input was true.

   ```
   node once_since(c, a : bool) returns (o : bool);
   -- make clean; make MAIN=once_since TRACE=trace3.txt
   ```

4. Any time A has occurred in the past, either B has been continuously true, or C has occurred at least once, since the last occurrence of A

   ```
   node always_from_to(b, a, c : bool) returns (x : bool);
   -- make clean; make MAIN=always_from_to TRACE=trace4.txt
   ```