

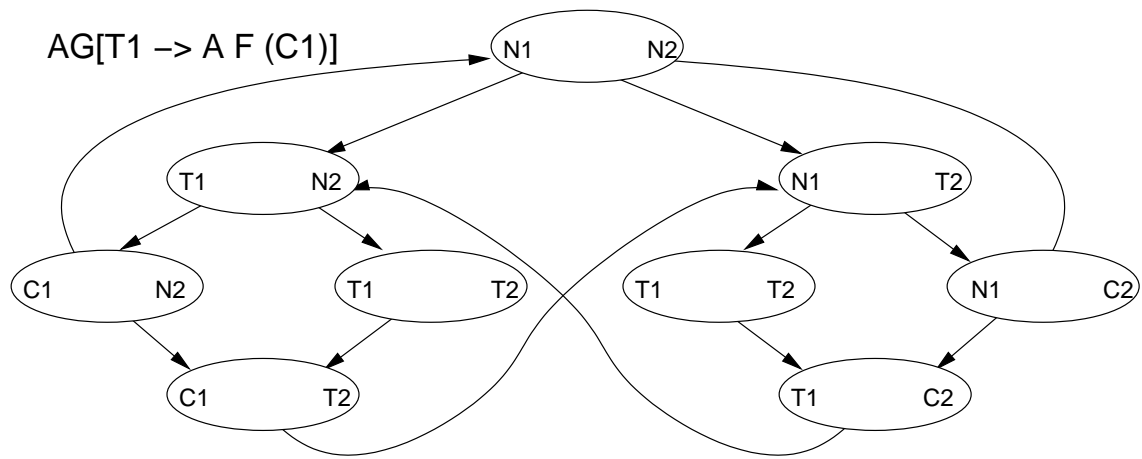
## Model checking by explicit state traversal

This example is taken from “Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specification,” by Clarke, Emerson, and Sistla (*ACM Transactions on Programming Languages and Systems*, **8**(2):244–263, April 1996). It is used to illustrate an algorithm showing that the complexity of checking CTL formula  $\phi$  in model  $M$  is  $O(\text{length}(\phi) \times (\#states(M) + \#edges(M)))$ .

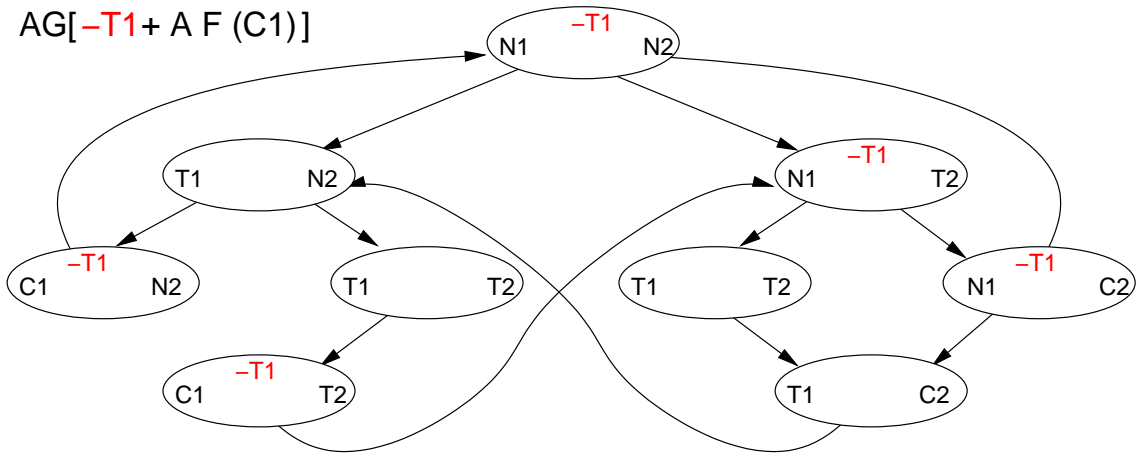
The algorithm proceeds bottom-up, according to the phrases of  $\phi$ , making one pass for each operator and quantifier. Each pass involves a recursive, depth-first traversal of  $M$  whose details depend on the operation or quantifier being evaluated. For example, the procedure for  $AF[\phi]$  is:

```
boolean procedure  $AF(\phi, s, M)$ 
  begin{Each state  $s \in M$  in which  $\phi$  is true has
    already been labeled with  $\phi$ .}
  if  $marked?(s)$  then return ( $AF[\phi] \in labels(s)$ );
   $mark(s, M)$  ;
  if  $\phi \in labels(s)$  then
    begin
       $add-label('AF[\phi]', labels(s))$ ;
      return true
    end
  else
    begin
      for each  $s' \in children(s)$  do
        if  $\neg AF(\phi, s', M)$  then return false;
       $add-label('AF[\phi]', labels(s))$ ;
      return true
    end
  end
```

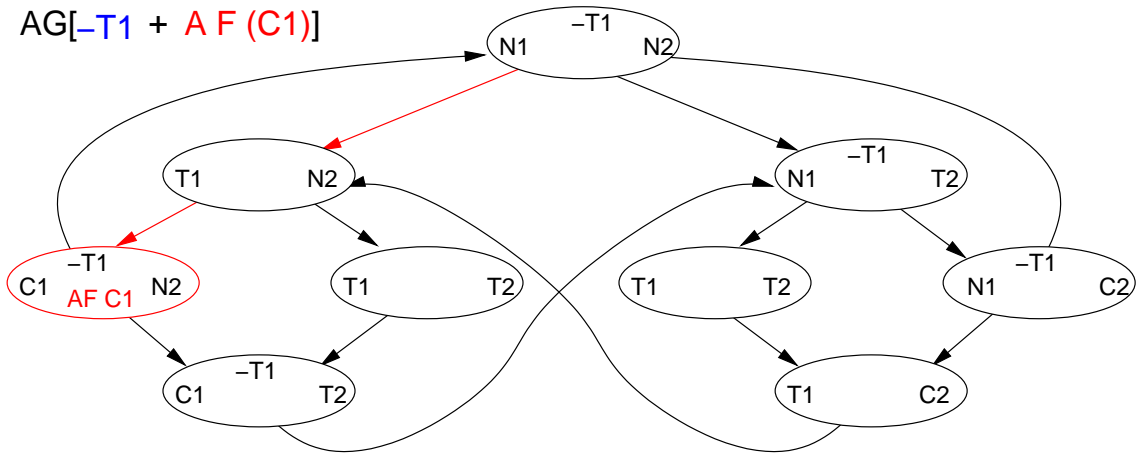
The following diagrams illustrate the evaluation of  $AG[T1 \rightarrow AF[C_1]]$  for the mutual exclusion model.



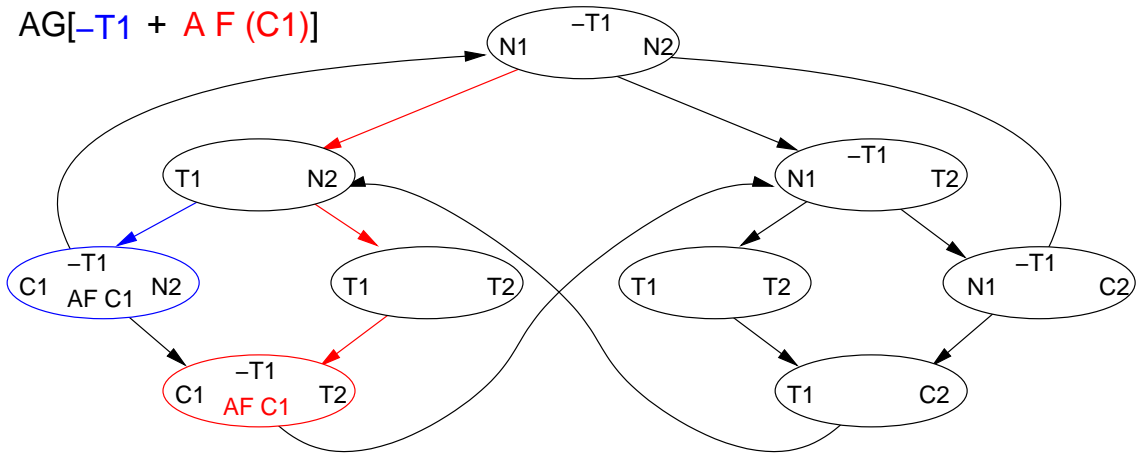
AG[-T1 + A F (C1)]



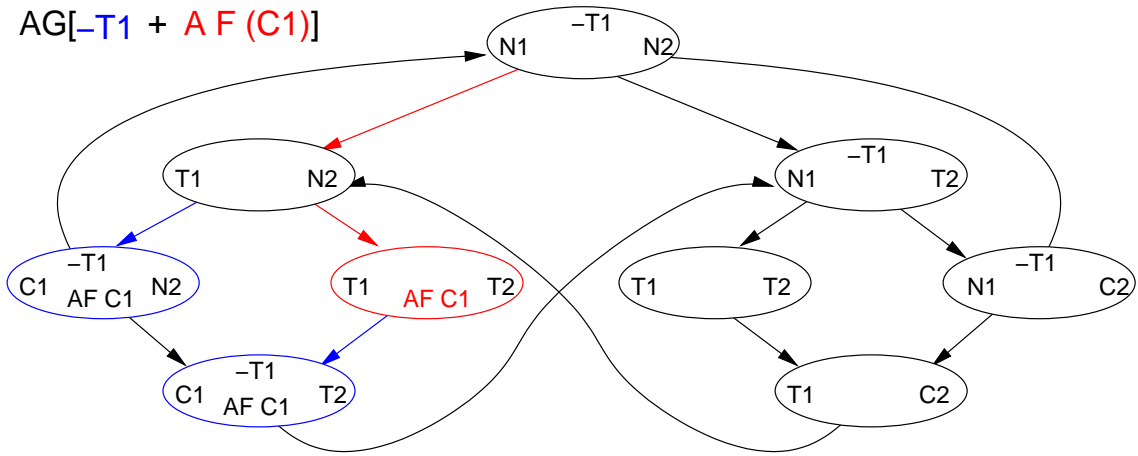
AG[-T1 + AF(C1)]



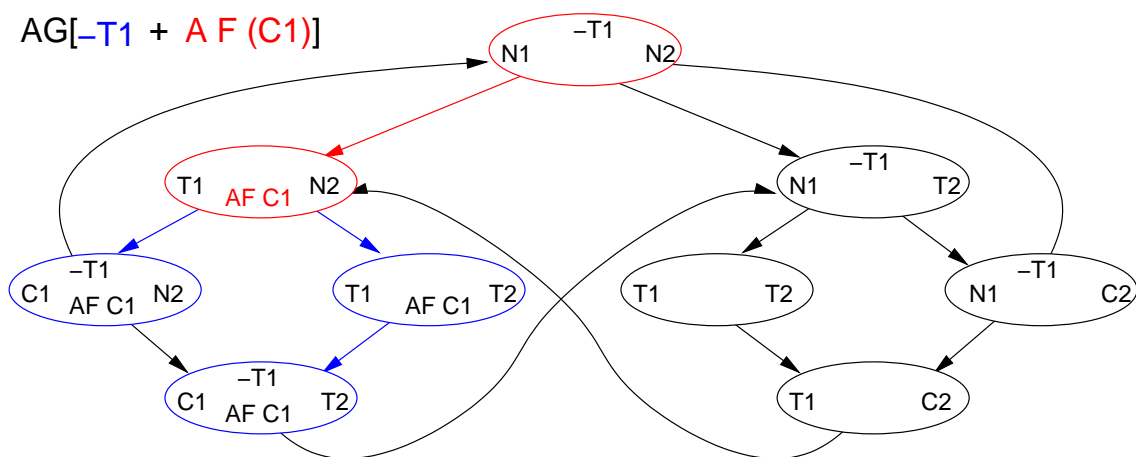
AG[-T1 + AF(C1)]



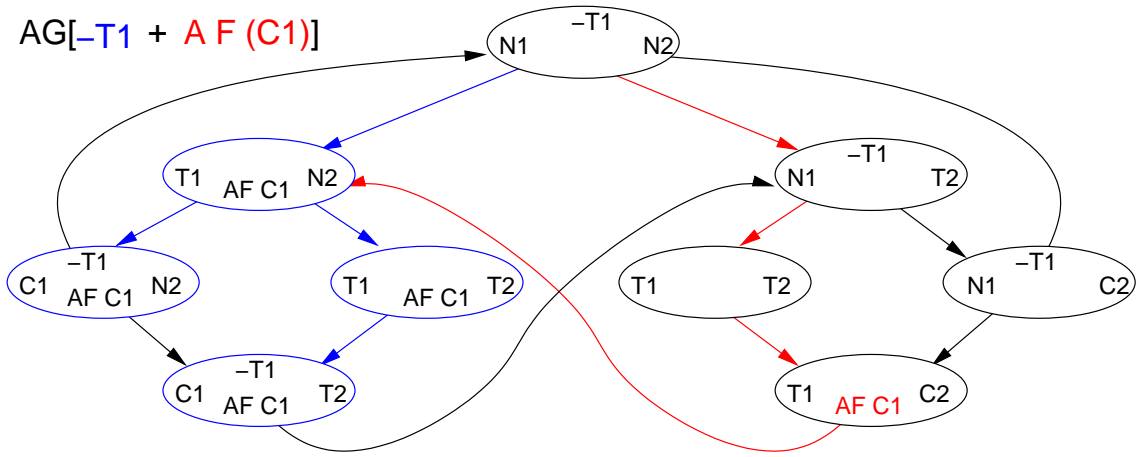
AG[-T1 + AF(C1)]



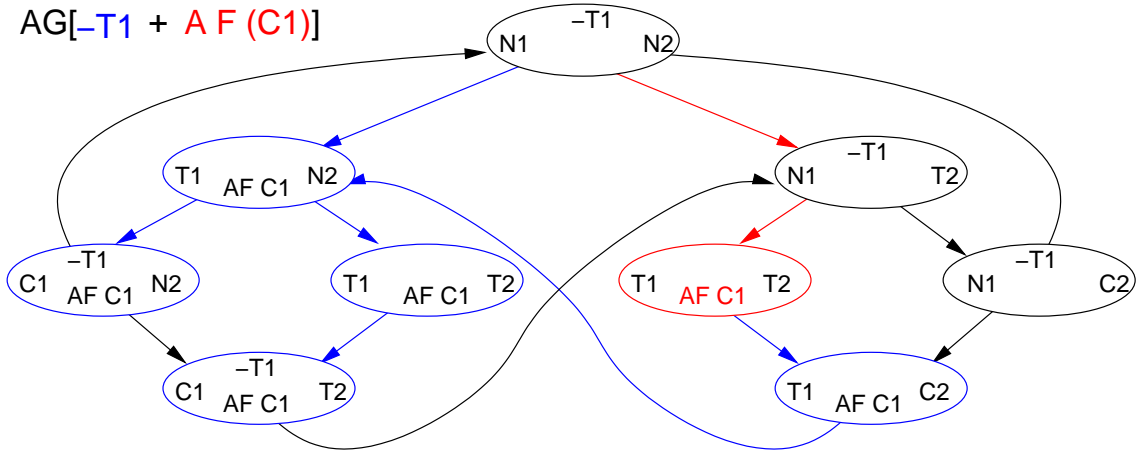
AG[-T1 + AF(C1)]



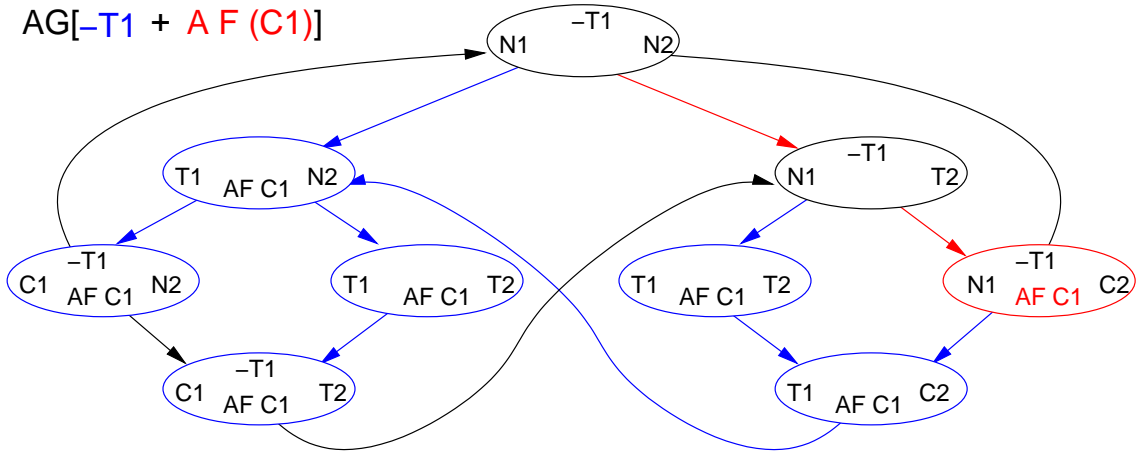
AG[-T1 + AF(C1)]



AG[-T1 + AF(C1)]

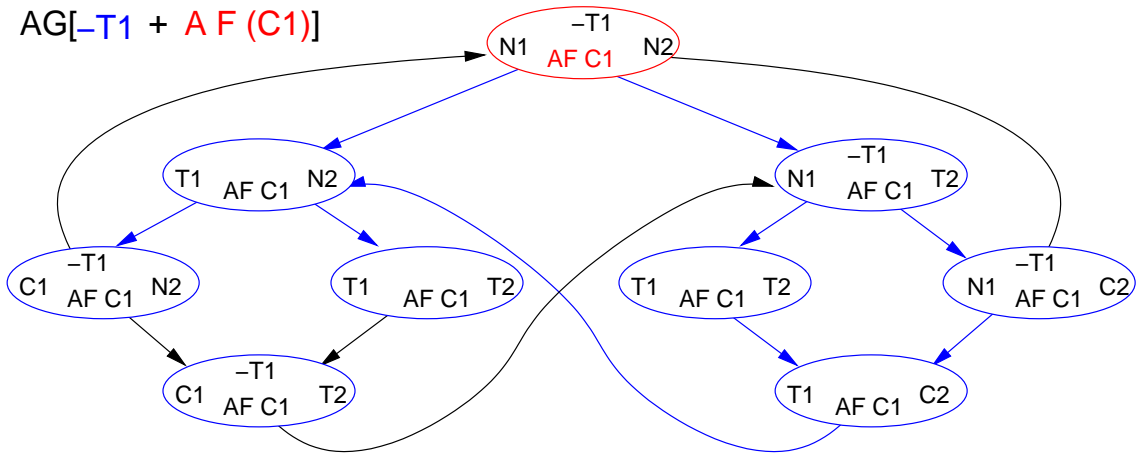


AG[-T1 + AF(C1)]

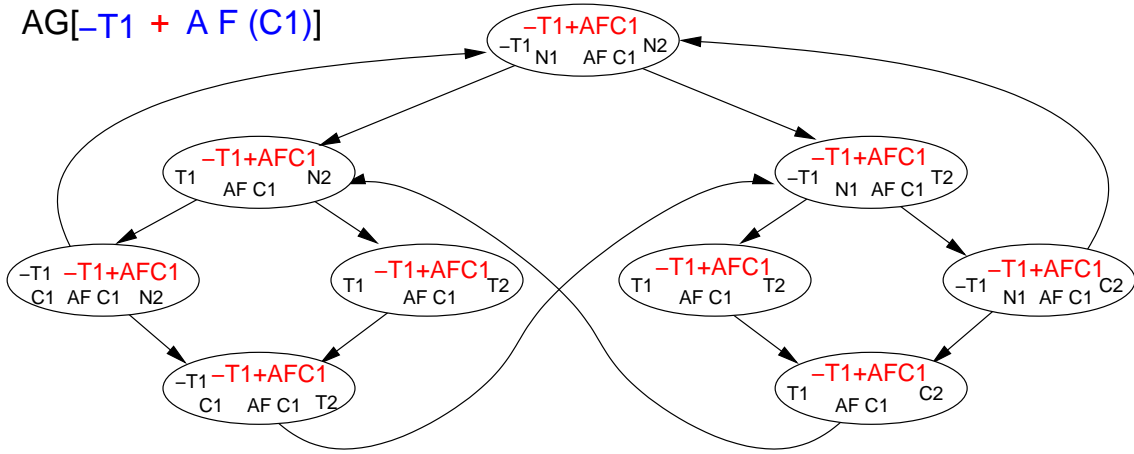




AG[-T1 + AF(C1)]



AG[-T1 + A F (C1)]



AG[-T1 + A F (C1)]

