procedure *CheckEG*(f_1)
    $S' := \{ s \mid f_1 \in label(s) \}$;
    $SCC := \{ C \mid C$ is a nontrivial SCC of $S' \}$;
    $T := \bigcup_{C \in SCC} \{ s \mid s \in C \}$;
    for all $s \in T$ **do** $label(s) := label(s) \cup \{ \mathbf{EG}\ f_1 \}$;
    while $T \neq \emptyset$ **do**
        choose $s \in T$;
        $T := T \setminus \{s\}$;
        for all t **such that** $t \in S'$ **and** $R(t, s)$ **do**
            if $\mathbf{EG}\ f_1 \notin label(t)$ **then**
                $label(t) := label(t) \cup \{ \mathbf{EG}\ f_1 \}$;
                $T := T \cup \{t\}$;
            end if;
        end for all;
    end while;
end procedure

Figure 4.2
Procedure for labeling the states satisfying **EG** f_1.

is a path within C between any pair of states in C. Let s_1 and s_2 be states in C. Pick some instance of s_1 on π_1. By the way in which π_1 was selected, we know that there is an instance of s_2 further along π_1. The segment from s_1 to s_2 lies entirely within C. This segment is a finite path from s_1 to s_2 in C. Thus, either C is a strongly connected component or it is contained within one. In either case, both conditions (1) and (2) are satisfied.

Next, assume that Conditions (1) and (2) are satisfied. Let π_0 be the path from s to t. Let π_1 be a finite path of length at least 1 that leads from t back to t. The existence of π_1 is guaranteed because t is a state in a nontrivial strongly connected component. All the states on the infinite path $\pi = \pi_0 \pi_1^{\omega}$ satisfy f_1. Since π is also a possible path starting at s in M, we see that $M, s \models \mathbf{EG}\ f_1$. $\square$

The algorithm for the case of $g = \mathbf{EG}\ f_1$ follows directly from the lemma. We construct the restricted Kripke structure $M' = (S', R', L')$ as described above. We partition the graph (S', R') into strongly connected components using the algorithm of Tarjan [2]. This algorithm has time complexity $O(|S'| + |R'|)$. Next, we find those states that belong to nontrivial components. We then work backward using the converse of R' and find all of those states that can be reached by a path in which each state is labeled with f_1. The entire computation can be performed in time $O(|S| + |R|)$. In Figure 4.2 we give a procedure