4 Model Checking

The *model checking problem* is easy to describe. Given a Kripke structure $M = (S, R, L)$ that represents a finite-state concurrent system and a temporal logic formula f expressing some desired specification, find the set of all states in S that satisfy f:

$$\{ s \in S \mid M, s \models f \}.$$

Normally, some states of the concurrent system are designated as *initial states*. The system satisfies the specification provided that all of the initial states are in the set.

The first algorithms for solving the model checking problem used an *explicit* representation of the Kripke structure as a labeled, directed graph with arcs given by pointers. In this case, the nodes represent the states in S, the arcs in the graph give the transition relation R, and the labels associated with the nodes describe the function $L : S \to 2^{AP}$.

4.1 CTL Model Checking

Let $M = (S, R, L)$ be a Kripke structure. Assume that we want to determine which states in S satisfy the CTL formula f. The algorithm will operate by labeling each state s with the set *label*(s) of subformulas of f which are true in s. Initially, *label*(s) is just $L(s)$. The algorithm then goes through a series of stages. During the ith stage, subformulas with $i - 1$ nested CTL operators are processed. When a subformula is processed, it is added to the labeling of each state in which it is true. Once the algorithm terminates, we will have that $M, s \models f$ iff $f \in$ *label*(s).

Recall that any CTL formula can be expressed in terms of $\neg$, $\vee$, **EX**, **EU** and **EG**. Thus, for the intermediate stages of the algorithm it is sufficient to be able to handle six cases, depending on whether g is atomic or has one of the following forms: $\neg f_1$, $f_1 \vee f_2$, $\mathbf{EX}\ f_1$, $\mathbf{E}[f_1\ \mathbf{U}\ f_2]$, or $\mathbf{EG}\ f_1$.

For formulas of the form $\neg f_1$, we label those states that are not labeled by f_1. For $f_1 \vee f_2$, we label any state that is labeled either by f_1 or by f_2. For $\mathbf{EX}\ f_1$, we label every state that has some successor labeled by f_1.

To handle formulas of the form $g = \mathbf{E}[f_1\ \mathbf{U}\ f_2]$ we first find all states that are labeled with f_2. We then work backwards using the converse of the transition relation R and find all states that can be reached by a path in which each state is labeled with f_1. All such states should be labeled with g.

In Figure 4.1 we give a procedure $CheckEU$ that adds $\mathbf{E}[f_1\ \mathbf{U}\ f_2]$ to *label*(s) for every s that satisfies $\mathbf{E}[f_1\ \mathbf{U}\ f_2]$, assuming that f_1 and f_2 have already been processed correctly, that is, for every state s, $f_1 \in$ *label*(s) iff $s \models f_1$ and $f_2 \in$ *label*(s) iff $s \models f_2$. This procedure requires time $O(|\ S\ | + |\ R\ |)$.