CTL Model Checking Algorithms

Formal Methods – Unit V

-Dr. Varsha Singh

Lecture Objectives

- Understand the syntax and semantics of CTL
- Learn the model checking algorithm for CTL
- Apply labeling algorithm on Kripke structures
- Analyze complexity and correctness of CTL checking

Recap: What is CTL?

- CTL = Computation Tree Logic
- Allows branching time temporal logic
- Formula: combines path quantifiers (A/E) with temporal operators (X, F, G, U)

CTL Syntax & Operators

- Path quantifiers: A (for all paths), E (exists a path)
- Temporal operators: X (next), F (eventually), G (globally), U (until)
- Examples: AF p, EG q, E[p U r]

Kripke Structure Refresher

- States, Transitions, Atomic Propositions
- Used as input model for CTL checking
- Labeled with sets of atomic propositions

Key Idea of CTL Model Checking

- Recursively evaluate formula bottom-up
- Label states satisfying sub-formulas
- Final label shows whether the initial state satisfies the CTL formula

Labeling Algorithm – Overview

Steps:

1. Parse formula into subformulas
2. For each subformula, compute set of satisfying states
3. Label Kripke states accordingly
4. Check if initial state is labeled with the formula

Atomic Propositions and Boolean Connectives

- Base case: Label states where atomic propositions hold
- Negation: complement the label set
- Conjunction: intersect label sets

Temporal Operators – EX, EU, EG

- EX p: A state has a successor labeled with p
- E[p U q]: Compute least fixpoint until q holds
- EG p: Compute greatest fixpoint where p holds globally

Example: Model Checking EG p

Steps:

1. Start with all states
2. Remove states not satisfying p or not closed under EG
3. Repeat until fixed point is reached

Example Walkthrough – E[p U q]

Steps:

1. Initialize with states satisfying q

2. Iteratively add states satisfying p with successor in the set

3. Stop when no new states can be added

Algorithm Termination & Complexity

- Fixpoint computations always terminate
- Each step: polynomial in size of model and formula
- Total complexity: $O(|M| \times |\Phi|)$

Applications of CTL Model Checking

- Hardware verification
- Protocol verification
- Safety and liveness checking

Summary

- CTL uses path quantifiers and temporal operators
- Model checking algorithm is recursive and bottom-up
- Fixpoint computations for temporal operators
- Efficient and automatable

Questions

- How does CTL differ from LTL in expressiveness?
- Why use fixpoints for EG and EU?
- Can we optimize model checking further?