The Fixed Point Characterization of CTL

Formal Methods – Unit V

Dr. Varsha Singh

Lecture Objectives

- Understand fixpoint theory in the context of temporal logic
- Explore least and greatest fixpoints
- Learn how CTL operators can be expressed using fixpoints
- Apply fixpoint characterization to model checking

Introduction to Fixpoints

- A fixpoint of a function f is a value x such that $f(x) = x$
- Important in defining recursive properties in logic and computation

Types of Fixpoints

- Least Fixpoint (LFP):

  The smallest solution x such that f(x) = x
- Greatest Fixpoint (GFP):

  The largest solution x such that f(x) = x
- Used respectively in 'until' and 'globally' CTL operators

Why Fixpoints in CTL?

- CTL operators like EU (exists until) and EG (exists globally) are inherently recursive
- Fixpoints provide a clean, formal way to define their semantics

Fixed Point Characterization – EU

- $E[p \; U \; q]$ = least fixpoint of: $\lambda Z.\; q \vee (p \wedge EX\; Z)$
- Start with states satisfying q and grow backward using p and EX

Fixed Point Characterization – EG

- EG p = greatest fixpoint of: $\lambda Z.\ p \wedge EX\ Z$
- Start with all states and eliminate those not satisfying the condition

Example: E[p U q]

1. Initialize Z0 = {s | q holds in s}
2. Repeat Z(i+1) = Z(i) ∪ {s | p holds in s and EX Z(i)}
3. Stop when Z(i+1) = Z(i) (convergence reached)

Example: EG p

1. Initialize $Z0 = \{s \mid p \text{ holds in } s\}$
2. Repeat $Z(i+1) = Z(i) \cap \{s \mid \exists \text{ successor in } Z(i)\}$
3. Continue until $Z(i+1) = Z(i)$

Visualizing Fixpoint Computation

- Iterative marking of states in the Kripke structure
- Convergence indicates property satisfaction in initial state

Relation to Model Checking

- Labeling algorithm is built on fixpoint evaluation
- Efficient and forms basis of many tools (e.g., NuSMV, SPIN)

Fixpoints and Modal Mu Calculus

- CTL is a fragment of modal μ-calculus
- μ-calculus uses fixpoint operators μ (least) and ν (greatest) explicitly

Strengths of Fixpoint View

- Mathematically precise and elegant
- Supports compositional reasoning
- Used in advanced verification tools

Summary

- Fixpoints define semantics of recursive temporal operators
- EU → least fixpoint, EG → greatest fixpoint
- Efficient fixpoint algorithms enable model checking
- Foundation for advanced logic like μ-calculus

Questions

- How does the fixpoint process guarantee termination?
- Can all CTL properties be expressed using fixpoints?
- Why are some fixpoints least and others greatest?