Lecture 2: CTL Model Checking

What is model checking?

State transition systems

Computation Tree Logics

The logic CTL

Typical CTL formulas

Structure of the SMV model checker

SMV examples

What is model checking?
Temporal Logic Model Checking

Specification Language:
A propositional temporal logic.

Verification Procedure:
Exhaustive search of the state space of the concurrent system to determine truth of specification.


Temporal Logic

(Unwind State Graph to obtain Infinite Tree)

Initial Computation Tree

Kripke Model

State Transition Graph or

Infinite Computation Tree
Computation Tree Logics

Formulas are constructed from path quantifiers and temporal operators:

1. Path quantifier:
   - \(\forall\) — for every path
   - \(\exists\) — there exists a path

2. Temporal Operator:
   - \(X\) — holds next time.
   - \(F\) — holds sometime in the future
   - \(G\) — holds globally in the future
   - \(U\) — holds until \(b\) holds

This lecture will deal primarily with CTL. The four most widely used CTL operators are illustrated below. (\$s_0$ is the root of each computation tree.)

The Logic CTL
• $\text{AG(}\text{EF} \text{ Restart})$: from any state it is possible to get to the Restart state.

• $\text{AG(}\text{AF DeviceEnabled})$: DeviceEnabled holds infinitely often on every computation path.

• $\text{AG(}\text{AF Ready Ack})$: if a Request occurs, then it will be eventually Acknowledged.

• $\text{AG(}\text{AF Ready} \leftrightarrow \text{AF Ack})$: if a Request occurs, then it will be eventually Ready does not hold.

• $\text{EF(}\text{Started} \lor \text{Ready})$: it is possible to get to a state where Started holds but Ready does not hold.

---

**Typical CTL Formulas**
Let \( M \) be the state–transition graph obtained from the concurrent system.

Let \( f \) be the specification expressed in temporal logic.

Find all states \( s \) of \( M \) such that
\[
\models f, s
\]

and check if initial states are among these.

Efficient model checking algorithms exist for CTL.

Symbolic Model Checking

Method used by most "industrial strength" model checkers:

- BDDs, traditionally used to represent boolean functions.
- Can handle much larger designs – hundreds of state variables.
- Uses boolean encoding for state machine and sets of states.
Model Checker Structure

Preprocessor

Temporal Logic Specification

True or Counterexample

Symbolic Model Verifier (SMV)
MODULE main

VAR
  request : boolean;
  state : {ready, busy};

ASSIGN
  init(state) := ready;

next(state) := case
  state = ready & request : busy;
end;

SPEC

AG(request -> AF state = busy)

VAR
  request : boolean;

MODULE main

A Simple SMV Example
MODULE main
VAR
    bit0: counter_cell(1);
    bit1: counter_cell(bit0.carry_out);
    bit2: counter_cell(bit1.carry_out);

SPEC
AAG !bit2, carry-out

VAR
MODULE counter-cell (carry-in)

VAR
MODULE main

A Three Bit Counter
Inverter Ring