COL750: Foundations of Automatic Verification (Jan-May 2023)

Lectures 09 & 10 (CTL Model Checking [with fairness] using BDDs)

Kumar Madhukar

madhukar@cse.iitd.ac.in

Feb 2nd and 13th

Existential Normal Form for CTL

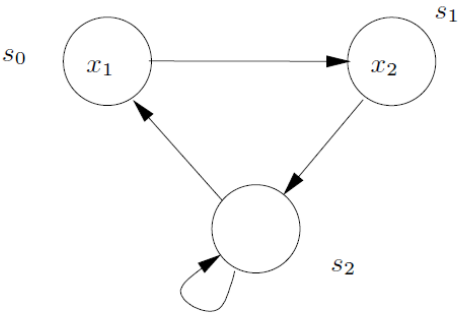
- $\phi := \mathit{true} \mid p_i \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid EX\phi \mid E(\phi_1 U \phi_2) \mid EG\phi$
- For every CTL formula there exists an equivalent CTL formula in ENF

- Correctness and Termination
- Efficiency

CTL Model Checking using BDDs

- encoding subsets of a finite set as OBDDs (characteristic function, using a "long enough" vector of booleans)
- for a set of states of $\mathcal{M} = (S, \rightarrow, L)$, there is a natural encoding (given by the labelling function)

Example



Representing subsets of states

set of states	representation by boolean values	representation by boolean function
$\emptyset$		0
$\{s_0\}$	$(1, 0)$	$x_1 \cdot \overline{x_2}$
$\{s_1\}$	$(0, 1)$	$\overline{x_1} \cdot x_2$
$\{s_2\}$	$(0, 0)$	$\overline{x_1} \cdot \overline{x_2}$
$\{s_0, s_1\}$	$(1, 0)$, $(0, 1)$	$x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2$
$\{s_0, s_2\}$	$(1, 0)$, $(0, 0)$	$x_1 \cdot \overline{x_2} + \overline{x_1} \cdot \overline{x_2}$
$\{s_1, s_2\}$	$(0, 1)$, $(0, 0)$	$\overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$
S	$(1, 0)$, $(0, 1)$, $(0, 0)$	$x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$

Representing the transition relation

- truth table with primed variables for next states

Representing the transition relation

- truth table with primed variables for next states
- interleaving unprimed and primed variables is usually more efficient

Representing the transition relation

- truth table with primed variables for next states
- interleaving unprimed and primed variables is usually more efficient
- implementation of $\mathit{pre}_{\exists}$

Representing the transition relation

- truth table with primed variables for next states
- interleaving unprimed and primed variables is usually more efficient
- implementation of $\mathit{pre}_{\exists}$
- not very useful to do this *via* truth tables

Synthesizing OBDDs from (compact) system descriptions

Exercise: Encode the model shown a couple of slides back in SMV, and extract the BDD for the transition relation from the SMV code (without creating a truth table explictly).

CTL Model Checking with Fairness

- recall the mutex example, where processes were allowed to stay in their critical section as long as required
- this can lead to violation of the liveness constraint AG $(t_1 \rightarrow \text{AF } c_1)$
- we would like to ignore such paths (assuming that the processes would eventually exit from its critical section after some finite time)
- In LTL, we could handle this by saying GF $\neg c_2 \rightarrow \phi$

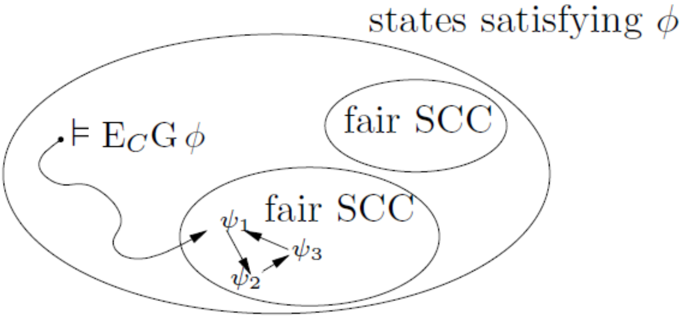
CTL Model Checking with Fairness

- CTL does not allow us to pick fair paths
- NuSMV allowed us to write FAIRNESS constraints
- NuSMV can handle only simple fairness constraints (of the form: ϕ is true infinitely often)
- fairness constraints may be more complex (e.g. if ϕ is true infinitely often, then ψ is true infinitely often)

Handling Simple Fairness

- Let $C := \{\psi_1, \psi_2, \ldots, \psi_n\}$ be n fairness constraints
- A computational path is called fair wrt these if every ψ_i is true infinitely often along that path
- Let A_C and E_C denote the operators A and E restricted to fair paths
- $E_C U$, $E_C X$, and $E_C G$ form an adequate set
- We need to handle only $E_C G$

Handing $E_C G$



states satisfying ϕ
$\models E_C G \phi$
fair SCC
fair SCC
ψ_1
ψ_2
ψ_3

Thank you!