CTL MC with Fairness

Barbara Jobstmann
CNRS/Verimag
Recall: CTL

• Syntax
  - $\neg \varphi$, $\varphi \land \psi$, $\varphi \lor \psi$, EX $\varphi$, E$\varphi$U$\psi$, EG$\varphi$

• Semantics:
  - $K,s \models \text{EX } \varphi$ iff $\exists$ a path $s_0,s_1,\ldots$ in $K$ s.t. $s=s_0$ and $s_1 \models \varphi$
  - $K,s \models \text{EG } \varphi$ iff $\exists$ a path $s_0,s_1,\ldots$ in $K$ s.t. $s=s_0$ and $\forall i \geq 0$, $K,si \models \varphi$ holds
  - $K,s \models \text{E} \varphi \text{U} \psi$ iff $\exists$ a path $s_0,s_1,\ldots$ in $K$ s.t. $s=s_0$ and $\exists i \geq 0$ with $K,si \models \varphi$ and $\forall 0 \leq j < i$, $K,sj \models \varphi$ holds
Recall: Symbolic Encoding

• Encoding:
  – state variables \( X \)
  – next-state variables \( X' \)
  – Set of states = predicate over \( X \)
  – Set of transitions = predicate over \( X \) and \( X' \)

• Representation of a system
  – Initial predicate \( I(X) \)
  – Transition predicate \( T(X,X') \)
Recall: Symbolic Computation

• MC Operations:
  - Union $P \cup Q$
  - Intersection $P \cap Q$
  - Complement $S \setminus P$
  - Element $s \in S$
  - Predecessor $\text{Pre}(S)$
  - Fix-points $\mu, \nu$

• Characteristic fcts
  - $p(x) \lor q(x)$
  - $p(x) \land q(x)$
  - $\neg p(x)$
  - $p(x_s) = 1$
  - $\exists x'. T(x, x') \land p(x')$
  - Repeated computation (Kleene’s theorem)
Recall: Fix-point Formulas

- Symbolic CTL Model Checking:
  - $\text{EF } \varphi = \mu Z. \varphi \lor \text{EX } Z$
  - $\text{E } \psi \text{ U } \varphi = \mu Z. \varphi \lor (\psi \land \text{EX } Z)$
  - $\text{EG } \varphi = \nu Z. \varphi \land \text{EX } Z$
Fairness Constraints

- Used to express constraints about environment
Example

- Dining Philosophers [Dijkstra'68]
  - allegory on process synchronization and resource sharing
  - 4 philosophers + 1 chop stick between each pair
  - Thinking, Hungry, Eating
  - 2 sticks need for eating
  - philosophers grab stick on left
  - philosophers ask for stick on right

Barbara Jobstmann
typedef enum {THINKING, HUNGRY, EATING} PhilState;

module Dining (clock, coins);
  input clock;
  input [3:0] coins;

  PhilState wire s0, s1, s2, s3;

  philosopher ph0 (clock, coins[0], s0, s1, s3);
  philosopher ph1 (clock, coins[1], s1, s2, s0);
  philosopher ph2 (clock, coins[2], s2, s3, s1);
  philosopher ph3 (clock, coins[3], s3, s0, s2);
endmodule // Dining

module philosopher (clock, coin, self, left, right);
  input clock;
  input coin;
  input left, right;
  output self;
  PhilState wire left;
  PhilState wire right;
  PhilState reg self;

  initial self = THINKING;

  always @(posedge clock) begin
    case (self)
      THINKING:
        if (coin) self = HUNGRY;
      HUNGRY:
        if (left != EATING &&
            right != EATING &&
            right != HUNGRY)
          self = EATING;
      EATING:
        if (coin) self = THINKING;
    endcase
  end
end // always @(posedge clock)
endmodule // philosopher
Problem?

• Progress? Deadlock?

\[ AG((s0=\text{HUNGRY} + s1=\text{HUNGRY} + s2=\text{HUNGRY} + s3=\text{HUNGRY}) \rightarrow AF(s0=\text{EATING} + s1=\text{EATING} + s2=\text{EATING} + s3=\text{EATING})) \];

• Let's add a book
typedef enum
{THINKING,HUNGRY,EATING, READING} PhilState;

module Dining(clock, coins);
  input clock;
  input [3:0] coins;
  PhilState wire  s0,s1,s2,s3;

philosopher ph0(clock,coins[0],s0,s1,s3,READING);
philosopher ph1(clock,coins[1],s1,s2,s0,THINKING);
philosopher ph2(clock,coins[2],s2,s3,s1,THINKING);
philosopher ph3(clock,coins[3],s3,s0,s2,THINKING);
endmodule // Dining

module philosopher(clock,coin,self,left,right, init);
  input clock,coin,left,right,init;
  output self;
  PhilState wire init,left,right;
  PhilState reg self;

  initial self = init;

  always @(posedge clock) begin
      case (self)
          THINKING:
              if (right == READING)
                  self = READING;
              else if (coin) self = HUNGRY;
          HUNGRY:
              if (left != EATING &&
                  right != EATING &&
                  right != HUNGRY)
                  self = EATING;
          EATING:
              if (coin) self = THINKING;
          READING:
              if (left == THINKING)
                  self = THINKING;
      endcase
  end // always @(posedge clock)
endmodule // philosopher
Problem?

• Progress/Deadlock?
• Starvation?
  \[ AG((s0=HUNGRY) \rightarrow AF(s0=EATING)); \]
• Why?

Barbara Jobstmann
Problem?

- Progress/Deadlock?
- Starvation?
  \[ AG((s_0=\text{HUNGRY}) \rightarrow AF(s_0=\text{EATING})) \];
- Why?
  - One Philosopher keeps eating forever
  - Not very reasonable to assume
- How can we rule out this behavior?
One Solution

• Bound “eating time”, i.e., time a philosopher stays in EATING state

• Disadvantages:
  – how to choose suitable upper bound? Proof will only hold for that particular bound, need additional reasoning
  – Keeping trace of eating time might increase complexity of model (MC harder and more difficult to read)
Alternative

- Fairness Constraints
- Tell the model checker to ignore all executions, where one philosopher eats forever
- Restriction on paths of models
- Must be done with caution to avoid masking of real bugs
Fairness Constraints

• Sets of states (constraint) that must occur infinitely often along a computation path to be considered: Büchi constraints

• Fairness constraints restrict the path quantifiers (E and A) to fair paths

• EFφ holds at state s only if there exists a fair path from s along which φ holds

• AGφ holds at s if φ holds in all states reachable from s along fair paths.
Example

• 2 Philosophers with 2 fairness constraints

1) !(s0=EATING)
2) !(s1=EATING)

Does

\[ AG((s_0=\text{HUNGRY}) \rightarrow AF(s_0=\text{EATING})) ; \]

holds with fairness constraints 1) and 2)?
Expressiveness

• Can we encode fairness into CTL?
  • Idea: write CTL formula stating that $\varphi$ holds infinitely often alone all paths $AG \ A F\varphi$

  • However,
    $$AG \ A F\varphi \rightarrow \psi$$

  trivially holds if any unfair path exists (holds vacuously, antecedent is false)
Expressiveness

• CTL talks about states, fairness talks about paths
• CTL may refer to paths of certain states
• Key: statements over paths must be immediately preceded by path quantifiers (A and E), which turns $\varphi$ into a formula over states
• Fairness not expressible in CTL
CTL with Fairness

• Fairness constraints/conditions:
  \[ C = \{ c_1, c_2, \ldots, c_n \} \] ... set of sets of states
• \( c_i \) ... constraint set = set of states
  – CTL formula can be used to describe a fair set, since it describes set of states
• Fair path: a path in the model along which each fairness condition holds infinitely often
• Fair states: states reachable along fair path
Computing Fair States

• Assume one fairness condition \( c \)
• Let \( S_F \) be set of fair states:
  – \( s \) in \( S_F \) must be starting point of a path that
    (1) goes through some state in \( c \) and
    (2) is entirely contained in \( S_F \).
  – Note that path has to be infinite, so at least of length one
• Fix-point formula:
  \[ S_F = \nu Z. \ EX( E Z U (Z \land c)) \]
Computing Fair States

• $S_F = \nu Z. \text{EX}(E Z U (Z \land c))$
• $Z0=S$
• $Z1=..$

$E S U S_{\land c}$
$\text{EX}(E S U S_{\land c})$

Barbara Jobstmann
Example

• Assume $c=\{4,8\}$
• $S_F = \nu Z. \text{EX}( E Z U (Z \land c))$

\[
\begin{align*}
Z_0 & = \{0 - 9\} \\
Z_1 & = \text{EX}(\mu Y.(Z_0 \land c) \lor (Z_0 \land \text{EX}(Y))) = \text{EX}(\{0 = 8\}) = \{0 - 7\} \\
Z_2 & = \text{EX}(\mu Y.((0 - 7) \land \{4, 8\}) \lor ((0 - 7) \land \text{EX}(Y))) \\
& = \text{EX}(\mu Y.(\{4\}) \lor ((0 - 7) \land \text{EX}(Y))) = \text{EX}(\{0 - 5\}) = \{1 - 5\} \\
Z_3 & = \text{EX}(\mu Y.((0 - 5) \land \{4, 8\}) \lor ((0 - 5) \land \text{EX}(Y))) = Z_2
\end{align*}
\]
Multiple Constraints

• $C = \{c_1, c_2, \ldots, c_n\}$

$$S_F = \forall Z. \text{EX} \land_{c \in C} (E \ Z \ U \ (Z \land c))$$

• Checking $E_C G \varphi = EG \varphi$ on fair paths

$$E_C G \varphi = \forall Z. \varphi \land \text{EX} \land_{c \in C} (E \ Z \ U \ (Z \land c))$$

$\varphi$ has to hold along a fair path
CTL MC with Fairness

• $E_C G \phi = \nu Z. \phi \land EX \land_{c \in C} (E Z U (Z \land c))$
• $E_C X \phi = ?$
• $E_C \phi U \psi = ?$
CTL MC with Fairness

- $E_C G\varphi = \nu Z. \varphi \land EX \land_{c \in C} (E Z U (Z \land c))$
- $E_C X\varphi = EX(\varphi \land S_F)$
- $E_C \varphi U\psi = E\varphi U(\psi \land S_F)$ with fair states
  
  $S_F = \nu Z. EX \land_{c \in C} (E Z U (Z \land c))$
Complexity

• Symbolic:
  – double nested fix-point => quadratic number of image-computations (instead of linear like CTL without fairness)
  – Complex algorithm is linear
    [Gentilini, Piazza, Policriti'02]
    but not necessarily better in practice

• Explicit:
  – set of fair states = states that reach an SCC that intersects all fair sets (same as CTL without f.)
Summary

• CTL MC
  – EF φ = μZ. φ ∨ EX Z
  – E ψ U φ = μZ. φ ∨ (ψ ∧ EX Z)
  – EG φ = νZ. φ ∧ EX Z

• CTL MC with Fairness
  – EC F φ = μZ. (φ ∧ SF) ∨ EX Z
  – EC ψ U φ = μZ. (φ ∧ SF) ∨ (ψ ∧ EX Z)
  – EC G φ = νZ. φ ∧ EX ∧_{c∈C} (EZ U (Z ∧ c))

  SF = νZ. EX ∧_{c∈C} (EZ U (Z ∧ c))
• $K,0 \models E\{c_1,c_2\} G \varphi$ with $c_1=\{6,10\}$ and $c_2=\{4,8\}$

- Choose some order on $c_1,c_2$, then go from 0 to first constraint, then continue to next
- 0 to 6: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 6$
- We choose 8: $6 \rightarrow 1 \rightarrow 2 \rightarrow 7 \rightarrow 8$ and try to close loop => we fail, we continue from 8
- We get 10: $8 \rightarrow 9 \rightarrow 10$ and try to close loop => we succeed: $10 \rightarrow 7 \rightarrow 8$
- Witness: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 7 \rightarrow 8$
CEX: Summary

- $K,s \models E_C G \varphi$
- $K,s \models E_C X \varphi$
- $K,s \models E_C \varphi U \psi$