liveness-to-safety and totalising a transition system

Enrico Magnago

University of Trento,
Fondazione Bruno Kessler

Objective
Encode the search for a fair path as a reachability problem.

How
Every solution [path] for the reachability problem should represent a fair lasso-shape in the original system.

Single fairness encoding

Fair transition system $M := \langle S, I, T, F \rangle$

- for $x \in S$, introduce a new symbol l_x.
- The assignment of every such symbol l_x is non-deterministically chosen in the initial state and then it never changes (FROZENVAR).
- The *loopback* state is $lback := \bigwedge_{x \in S} l\_x = x$.
- Boolean symbol in_l, initially false and $in\_l' = in\_l \vee lback'$.
- Boolean symbol $fair\_l$, initially false and $fair\_l' = fair\_l \vee (in\_l \wedge F)$

Find path starting from I that reaches $lback \wedge fair\_l$.

Single fairness encoding: example

Modulo 8 counter

```
MODULE main
  VAR
    b0 : boolean;
    b1 : boolean;
    b2 : boolean;

  ASSIGN
    init(b0) := FALSE;
    init(b1) := FALSE;
    init(b2) := FALSE;
    next(b0) := !b0;
    next(b1) := (!b0 & b1) | (b0 & !b1);
    next(b2) := ((b0 & b1) & !b2) | (!(b0 & b1) & b2);

  DEFINE out := toint(b0) + 2 * toint(b1) + 4 * toint(b2);

  LTLSPEC G F out != 2;
```

Single fairness encoding: solution

Add the following:

```
FROZENVAR
  l_b0 : boolean;
  l_b1 : boolean;
  l_b2 : boolean;
DEFINE out_lback := toint(l_b0) + 2 * toint(l_b1) + 4 * toint(l_b2);

DEFINE lback := l_b0 = b0 & l_b1 = b1 & l_b2 = b2;
DEFINE fair := out = 2;

VAR
  in_loop : boolean;
  fair_loop : boolean;

INIT !in_loop & !fair_loop;
TRANS next(in_loop) = in_loop | next(lback);
TRANS next(fair_loop) = fair_loop | (in_loop & fair);

INVARSPEC !(fair_loop & lback);
```

Multiple fairness conditions

Q: How can we deal with multiple fairness conditions?

Multiple fairness conditions

Q: How can we deal with multiple fairness conditions?

A1 Usual reduction to single fairness.
Requires to visit fairness conditions in a predefined order, can cause very long loops even when a shorter one exists.

Multiple fairness conditions

Q: How can we deal with multiple fairness conditions?

A1 Usual reduction to single fairness.
Requires to visit fairness conditions in a predefined order, can cause very long loops even when a shorter one exists.

A2 Add one boolean symbol $fair\_l_i$ for each fairness. Look for a path that reaches $l\_back \wedge \bigwedge_i fair\_l_i$

Multiple fairness conditions encoding: example

Modulo 8 counters

```
MODULE main
VAR
  c0 : counter();
  c1 : counter();

LTLSPEC (F G c0.out != 2) | (F G c1.out != 4);

MODULE counter
VAR
  b0 : boolean;
  b1 : boolean;
  b2 : boolean;

ASSIGN
  init(b0) := FALSE;
  init(b1) := FALSE;
  init(b2) := FALSE;
  next(b0) := !b0;
  next(b1) := (!b0 & b1) | (b0 & !b1);
  next(b2) := ((b0 & b1) & !b2) | (!(b0 & b1) & b2);

DEFINE out := toint(b0) + 2 * toint(b1) + 4 * toint(b2);
```

Multiple fairness conditions encoding: solution

See files in examples for both versions.

Make a transition system total

Q: Given a transition system $M \coloneqq \langle S, I, T \rangle$, can we define $M_t \coloneqq \langle S_t, I_t, T_t \rangle$, such that every path in M has a corresponding path in M_t and T_t is total?

Make a transition system total

Q: Given a transition system $M := \langle S, I, T \rangle$, can we define $M_t := \langle S_t, I_t, T_t \rangle$, such that every path in M has a corresponding path in M_t and T_t is total?

- $S_t := S \cap \{err\}$,
- $I_t := I \wedge \neg err$,
- $T_t := ((\neg err \wedge T) \rightarrow \neg err') \wedge (err \vee \neg T) \rightarrow err'$.

If $M_t \models (G \neg err) \wedge \phi$ then $M \models \phi$.

Q: What's the relation between the err states of M_t and the deadlocks of M?

Make a transition system total

Q: Given a transition system $M := \langle S, I, T\rangle$, can we define $M_t := \langle S_t, I_t, T_t\rangle$, such that every path in M has a corresponding path in M_t and T_t is total?

- $S_t := S \cap \{err\}$,
- $I_t := I \wedge \neg err$,
- $T_t := ((\neg err \wedge T) \rightarrow \neg err') \wedge (err \vee \neg T) \rightarrow err'$.

If $M_t \models (G\neg err) \wedge \phi$ then $M \models \phi$.

Q: What's the relation between the err states of M_t and the deadlocks of M?
We are adding a lot of transitions!!