Verification

Lecture 4

Bernd Finkbeiner

UNIVERSITÄT
DES
SAARLANDES

Plan for today

- CTL model checking
  - The basic algorithm
  - Fairness
  - Counterexamples and witnesses

CTL fairness constraints

- An unconditional CTL fairness constraint is a formula of the form:

$$ufair = \bigwedge_{0<i\leq k} \mathrm{G\,F}\,\Psi_i$$

- A weak CTL fairness constraint is a formula of the form:

$$wfair = \bigwedge_{0<i\leq k} (\mathrm{F\,G}\,\Phi_i \rightarrow \mathrm{G\,F}\,\Psi_i)$$

- A strong CTL fairness constraint is a formula of the form:

$$sfair = \bigwedge_{0<i\leq k} (\mathrm{G\,F}\,\Phi_i \rightarrow \mathrm{G\,F}\,\Psi_i)$$

where $\mathrm{G\,F}$ means "infinitely often", $\mathrm{F\,G}$ means "eventually forever". Φ_i and Ψ_i (for $0 < i \leq k$) are CTL-formulas over *AP*.

A CTL fairness assumption *fair* is a conjunction of CTL fairness constraints.

CTL fairness constraints

Note that unconditional and weak fairness constraints are special cases of strong fairness constraints:

- An unconditional CTL fairness constraint is a formula of the form:

$$ufair = \bigwedge_{0<i\leq k} \mathrm{G}\,\mathrm{F}\,\Psi_i \quad = \bigwedge_{0<i\leq k} (\mathrm{G}\,\mathrm{F}\,true \rightarrow \mathrm{G}\,\mathrm{F}\,\Psi_i)$$

- A weak CTL fairness constraint is a formula of the form:

$$wfair = \bigwedge_{0<i\leq k} (\mathrm{F}\,\mathrm{G}\,\Phi_i \rightarrow \mathrm{G}\,\mathrm{F}\,\Psi_i) = \bigwedge_{0<i\leq k} (\mathrm{G}\,\mathrm{F}\,true \rightarrow \mathrm{G}\,\mathrm{F}\,(\neg\Phi_i \vee \Psi_i))$$

- A strong CTL fairness constraint is a formula of the form:

$$sfair = \bigwedge_{0<i\leq k} (\mathrm{G}\,\mathrm{F}\,\Phi_i \rightarrow \mathrm{G}\,\mathrm{F}\,\Psi_i)$$

where Φ_i and Ψ_i (for $0 < i \leq k$) are CTL-formulas over *AP*.

Semantics of fair CTL

For CTL fairness assumption *fair*, relation $\models_{fair}$ is defined by:

$s \models_{fair} a$	iff	$a \in L(s)$
$s \models_{fair} \neg \Phi$	iff	$\neg (s \models_{fair} \Phi)$
$s \models_{fair} \Phi \vee \Psi$	iff	$(s \models_{fair} \Phi) \vee (s \models_{fair} \Psi)$
$s \models_{fair} \mathrm{E}\, \varphi$	iff	$\pi \models_{fair} \varphi$ for some fair path π that starts in s
$s \models_{fair} \mathrm{A}\, \varphi$	iff	$\pi \models_{fair} \varphi$ for all fair paths π that start in s

$\pi \models_{fair} \mathrm{X}\, \Phi$	iff	$\pi[1] \models_{fair} \Phi$
$\pi \models_{fair} \Phi\, \mathrm{U}\, \Psi$	iff	$(\exists j \geq 0.\ \pi[j] \models_{fair} \Psi \wedge (\forall\, 0 \leq k < j.\ \pi[k] \models_{fair} \Phi))$

π is a fair path iff $\pi \models$ *fair* for CTL fairness assumption *fair*

Transition system semantics

- For CTL-state-formula Φ, and fairness assumption *fair*, the satisfaction set $Sat_{fair}(\Phi)$ is defined by:

$$Sat_{fair}(\Phi) = \{ q \in S \mid q \models_{fair} \Phi \}$$

- *TS* satisfies CTL-formula Φ iff Φ holds in all its initial states:

$$TS \models_{fair} \Phi \quad \text{if and only if} \quad \forall q_0 \in I.\, q_0 \models_{fair} \Phi$$

  - this is equivalent to $I \subseteq Sat_{fair}(\Phi)$

Fair CTL model-checking problem

For:

- finite transition system
- CTL formula Φ in ENF, and
- CTL fairness assumption *fair*

establish whether or not:

$$TS \models_{fair} \Phi$$

use bottom-up procedure a la CTL to determine $Sat_{fair}(\Phi)$
using as much as possible standard CTL model-checking algorithms

CTL fairness constraints

- Let $sfair = \bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\,\Phi_i \rightarrow \mathsf{G}\,\mathsf{F}\,\Psi_i)$
  - where Φ_i and Ψ_i (for $0 < i \leq k$) are CTL-formulas over AP
- Replace the CTL state-formulas in *sfair* by fresh atomic propositions:

$$sfair := \bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\,a_i \rightarrow \mathsf{G}\,\mathsf{F}\,b_i)$$

  - where $a_i \in L(s)$ if and only if $s \in Sat(\Phi_i)$ (not $Sat_{fair}(\Phi_i)$!)
  - $b_i \in L(s)$ if and only if $s \in Sat(\Psi_i)$ (not $Sat_{fair}(\Psi_i)$!)

Results for $\vDash_{fair}$ (1)

$\pi \vDash fair$ iff $\pi[j..] \vDash fair$ for some $j \geq 0$ iff $\pi[j..] \vDash fair$ for all $j \geq 0$

- $s \vDash_{fair} \mathrm{EX}\, a$ if and only if $\exists s' \in Post(s)$ with $s' \vDash a$ and $FairPaths(s') \neq \varnothing$
- $s \vDash_{fair} \mathrm{E}\,(a \,\mathrm{U}\, a')$ if and only if there exists a finite path fragment

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in Paths_{fin}(s) \qquad \text{with } n \geq 0$$

  such that $s_i \vDash a$ for $0 \leq i < n$, $s_n \vDash a'$, and $FairPaths(s_n) \neq \varnothing$

Results for $\models_{fair}$ (2)

- $s \models_{fair} \mathrm{EX}\, a$ if and only if $\exists s' \in Post(s)$ with $s' \models a$ and $\underbrace{FairPaths(s') \neq \varnothing}_{s' \models_{fair} \mathrm{EG\,true}}$

- $s \models_{fair} \mathrm{E}\,(a \,\mathrm{U}\, a')$ if and only if there exists a finite path fragment

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in Paths_{fin}(s) \qquad \text{with } n \geq 0$$

such that $s_i \models a$ for $0 \leq i < n$, $s_n \models a'$, and $\underbrace{FairPaths(s_n) \neq \varnothing}_{s_n \models_{fair} \mathrm{EG\,true}}$

Basic algorithm

- Determine $Sat_{fair}(\text{EG true}) = \{ q \in S \mid FairPaths(q) \neq \varnothing \}$
- Introduce an atomic proposition a_{fair} such that:
  - $a_{fair} \in L(q)$ if and only if $q \in Sat_{fair}(\text{EG true})$
- Compute the sets $Sat_{fair}(\Psi)$ for all subformulas Ψ of Φ (in ENF)

by:

$$
\begin{aligned}
Sat_{fair}(a) &= \{ q \in S \mid a \in L(q) \} \\
Sat_{fair}(\neg a) &= S \setminus Sat_{fair}(a) \\
Sat_{fair}(a \wedge a') &= Sat_{fair}(a) \cap Sat_{fair}(a') \\
Sat_{fair}(\text{EX}\, a) &= Sat\,(\text{EX}\,(a \wedge a_{fair})) \\
Sat_{fair}(\text{E}\,(a\, \text{U}\, a')) &= Sat\,(\text{E}\,(a\, \text{U}\,(a' \wedge a_{fair}))) \\
Sat_{fair}(\text{EG}\, a) &= \ldots\ldots
\end{aligned}
$$

- Thus: model checking CTL under fairness constraints is
  - CTL model checking + algorithm for computing $Sat_{fair}(\text{EG}\, a)$!

Core model-checking algorithm

{states are assumed to be labeled with a_i and b_i}
compute $Sat_{fair}(\text{EG true}) = \{ q \in S \mid FairPaths(q) \neq \varnothing \}$
forall $q \in Sat_{fair}(\text{EG true})$ **do** $L(q) := L(q) \cup \{ a_{fair} \}$ **od**
{compute $Sat_{fair}(\Phi)$}
for all $0 < i \leq |\Phi|$ **do**
  for all $\Psi \in Sub(\Phi)$ with $|\Psi| = i$ **do**
    switch(Ψ):
      true : $Sat_{fair}(\Psi) := S;$
      a : $Sat_{fair}(\Psi) := \{ q \in S \mid a \in L(s) \};$
      $a \wedge a'$: $Sat_{fair}(\Psi) := \{ q \in S \mid a, a' \in L(s) \};$
      $\neg a$: $Sat_{fair}(\Psi) := \{ q \in S \mid a \notin L(s) \};$
      $\text{EX}\, a$: $Sat_{fair}(\Psi) := Sat(\text{EX}\,(a \wedge a_{fair}));$
      $\text{E}\,(a \,\text{U}\, a')$: $Sat_{fair}(\Psi) := Sat(\text{E}\,(a \,\text{U}\, (a' \wedge a_{fair})));$
      $\text{EG}\, a$: compute $Sat_{fair}(\text{EG}\, a)$
    end switch
    replace all occurrences of Ψ (in Φ) by the fresh atomic proposition a_Ψ
    forall $q \in Sat_{fair}(\Psi)$ **do** $L(q) := L(q) \cup \{ a_\Psi \}$ **od**
  end for
end for
return $I \subseteq Sat_{fair}(\Phi)$

Characterization of $Sat_{fair}(\mathrm{EG}\, a)$

$$q \vDash_{sfair} \mathrm{EG}\, a \quad \text{where} \quad sfair = \bigwedge_{0<i\leq k} (\mathrm{G\,F}\, a_i \rightarrow \mathrm{G\,F}\, b_i)$$

iff there exists a finite path fragment $q_0 \ldots q_n$ and a cycle $q'_0 \ldots q'_r$ with:

1. $q_0 = q$ and $q_n = q'_0 = q'_r$
2. $q_i \vDash a$, for any $0 \leq i \leq n$, and $q'_j \vDash a$, for any $0 \leq j \leq r$, and
3. $Sat(a_i) \cap \{ q'_1, \ldots, q'_r \} = \varnothing$ or $Sat(b_i) \cap \{ q'_1, \ldots, q'_r \} \neq \varnothing$ for $0 < i \leq k$

Computing $Sat_{fair}(\mathrm{EG}\, a)$

- Consider state q only if $q \models a$, otherwise eliminate q
  - change TS into $TS[a] = (S', Act, \rightarrow', I', AP, L')$ with $S' = Sat(a)$,
  - $\rightarrow' = \rightarrow \cap (S' \times Act \times S')$, $I' = I \cap S'$, and $L'(s) = L(s)$ for $s \in S'$

  $\Rightarrow$ each infinite path fragment in $TS[a]$ satisfies $\mathrm{G}\, a$
- $q \models_{fair} \mathrm{EG}\, a$ iff there is a non-trivial strongly connected set of nodes D in $TS[a]$ reachable from q such that
  - $D \cap Sat(a_i) = \varnothing$ or
  - $D \cap Sat(b_i) \neq \varnothing$

  for $0 < i \leq k$
- $Sat_{sfair}(\mathrm{EG}\, a) = \{ q \in S \mid Reach_{TS[a]}(s) \cap T \neq \varnothing \}$
  - T is the union of all such SCCs D.

how to compute T?

Unconditional fairness

$$ufair \equiv \bigwedge_{0<i\leq k} \mathrm{G}\,\mathrm{F}\, b_i$$

Let T be the set union of all non-trivial SCCs C of $TS[a]$ satisfying

$$C \cap Sat(b_i) \neq \varnothing \quad \text{for all } 0 < i \leq k$$

It now follows:

$$s \models_{ufair} \mathrm{EG}\, a \quad \text{if and only if} \quad Reach_{G[a]}(s) \cap T \neq \varnothing$$

$\Rightarrow$ T can be determined by a simple graph analysis (DFS)

Strong fairness: single constraint ($k = 1$)

- $sfair = \mathsf{G}\,\mathsf{F}\,a_1 \rightarrow \mathsf{G}\,\mathsf{F}\,b_1$
- $q \models_{sfair} \mathsf{E}\,\mathsf{G}\,a$ iff C is a non-trivial SCC in $TS[a]$ reachable from q with:
  (1) $C \cap Sat(b_1) \neq \varnothing$, or
  (2) there exists a non-trivial SCC D in $C[\neg a_1]$
- For the union T of all such SCCs C:

$$q \models_{sfair} \mathsf{E}\,\mathsf{G}\,a \quad \text{if and only if} \quad Reach_{S[a]}(q) \cap T \neq \varnothing$$

Strong fairness: general case ($k > 1$)

Check each non-trivial SCC C recursively as follows:

Check$(C, \bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\, a_i \rightarrow \mathsf{G}\,\mathsf{F}\, b_i))$:

if $\forall i \in \{1, \ldots, k\} : C \cap Sat(b_i) \neq \emptyset$ **return** *true*

else

choose some $j \in \{1, \ldots, k\} : C \cap Sat(b_j) = \emptyset$.
remove all states in $Sat(a_j)$ from C
for all non-trivial SCCs D do

if Check$(D, \bigwedge_{0<i\leq k, i\neq j} (\mathsf{G}\,\mathsf{F}\, a_i \rightarrow \mathsf{G}\,\mathsf{F}\, b_i))$ **return** *true*

return *false*

T is the union of all SCCs C that pass the check.

Complexity

For a transition system *TS* with *N* states and *M* transitions,

CTL formula Φ, and CTL fairness constraint *fair* with *k* conjuncts,

the CTL model-checking problem $TS \models_{fair} \Phi$

can be determined in time $\mathcal{O}(|\Phi| \cdot (N + M) \cdot k)$

Counterexamples and Witnesses

Counterexamples

- Model checking is an effective and efficient "bug hunting" technique
- Counterexamples are important for diagnostic feedback, abstraction-refinement, schedule synthesis . . .
- $TS \not\models \mathsf{A}\,\varphi$ where φ only contains universal path quantifiers
  - counterexample = a sufficiently long prefix of a path refuting φ
  - this fragment of the logic is known as universal fragment of CTL
- $TS \not\models \mathsf{E}\,\varphi$ where φ is arbitrary CTL formula
  - all paths satisfy $\neg\varphi$! $\Rightarrow$ no clear notion of counterexample
  - witness = a sufficiently long prefix of a path satisfying φ
- So:
  - for $\mathsf{A}\,\varphi$, a prefix of π with $\pi \not\models \varphi$ acts as counterexample
  - for $\mathsf{E}\,\varphi$, a prefix of π with $\pi \models \varphi$ acts as witness

Counterexamples for X Φ

- A counterexample of X Φ is a path fragment $q\,q'$ with
  - $q \in I$ and $q' \in Post(q)$ with $q' \not\models \Phi$
- A witness of X Φ is a is a path fragment $q\,q'$ with
  - $q \in I$ and $q' \in Post(q)$ with $q' \models \Phi$
- Algorithm: inspection of direct successors of initial states

Counterexamples for G Φ

- Counterexample is initial path fragment $q_0\, q_1 \ldots q_n$ such that:
  - $q_0, \ldots, q_{n-1} \vDash \Phi$ and $q_n \nvDash \Phi$
- Algorithm: backward search starting in $\neg\Phi$-states
- A witness of $\varphi = \mathrm{G}\,\Phi$ consists of an initial path fragment of the form:
  - $\underbrace{q_0\, q_1 \ldots q_n\, q'_1 \ldots q'_r}_{\text{satisfy } \Phi}$ with $q_n = q'_r$
- Algorithm: cycle search in the digraph $G = (S, E')$ where the set of edges E':
  - $E' = \{\, (q, q') \mid q' \in Post(q) \;\wedge\; q \vDash \Phi \,\}$

Counterexamples for $\Phi \, \mathsf{U} \, \Psi$

- A witness is an initial path fragment $q_0 \, q_1 \ldots q_n$ with
  - $q_n \models \Psi$ and $q_i \models \Phi$ for $0 \leq i < n$
- Algorithm: backward search starting in the set of Ψ-states
- A counterexample is an initial path fragment that indicates a path π:
  - for which either
    $\pi \models \mathsf{G}\,(\Phi \wedge \neg\Psi)$ or $\pi \models (\Phi \wedge \neg\Psi)\,\mathsf{U}\,(\neg\Phi \wedge \neg\Psi)$
- Counterexample is initial path fragment of either form:
  - $$\underbrace{q_0 \ldots q_{n-1} \, \underbrace{q_n \, q'_1 \ldots q'_r}_{\text{cycle}}}_{\text{satisfy } \Phi \wedge \neg\Psi} \quad \text{with } q_n = q'_r$$ or
    $$\underbrace{q_0 \ldots q_{n-1}}_{\text{satisfy } \Phi \wedge \neg\Psi} \quad q_n \quad \text{with } q_n \models \neg\Phi \wedge \neg\Psi$$

Counterexample generation

- Determine the SCCs of the digraph $G = (S, E')$ where

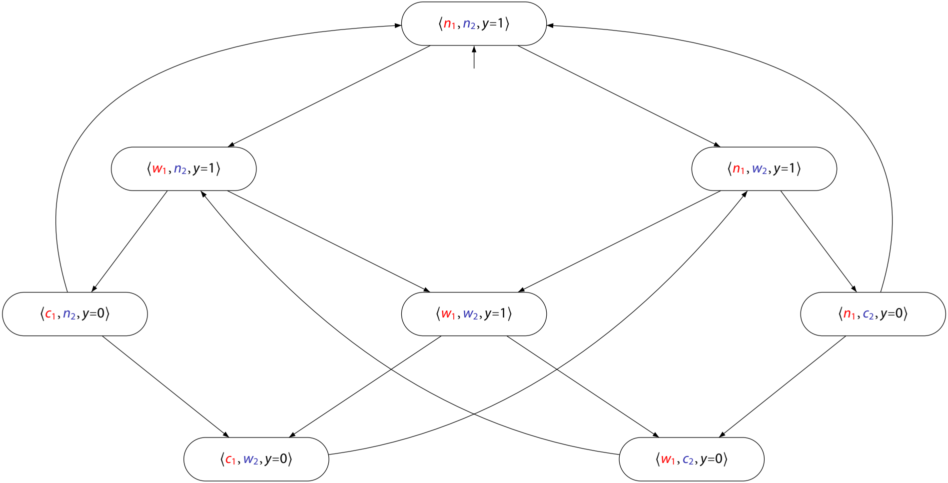
$$E' = \{ (q, q') \in S \times S \mid q' \in Post(q) \ \wedge \ q \models \Phi \ \wedge \ \neg\Psi \}$$

- Each path in G that starts in an initial state $q_0 \in I$ and leads to a non-trivial SCC C in G provides a counterexample of the form:

$$q_0 \, q_1 \ldots q_n \underbrace{q'_1 \ldots q'_r}_{\in C} \quad \text{with} \quad q_n = q'_r$$

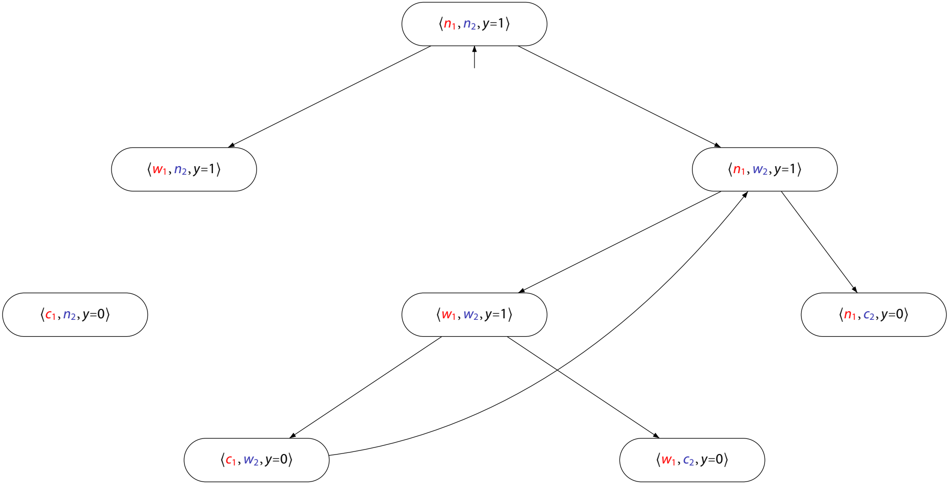
- Each path in G that leads from an initial state q_0 to a trivial terminal SCC $C = \{ q' \}$ with $q' \not\models \Psi$ provides a counterexample of the form $q_0 \, q_1 \ldots q_n$ with $q_n \models \neg\Phi \wedge \neg\Psi$

Example



$$\mathsf{A}\Big(\underbrace{((n_1 \wedge n_2) \vee w_2)}_{\Phi} \;\mathsf{U}\; \underbrace{c_2}_{\Psi}\Big)$$

SCC graph



$\langle n_1, n_2, y{=}1\rangle$
$\langle w_1, n_2, y{=}1\rangle$
$\langle n_1, w_2, y{=}1\rangle$
$\langle c_1, n_2, y{=}0\rangle$
$\langle w_1, w_2, y{=}1\rangle$
$\langle n_1, c_2, y{=}0\rangle$
$\langle c_1, w_2, y{=}0\rangle$
$\langle w_1, c_2, y{=}0\rangle$

Complexity

Let TS be a transition system TS with N states and M transitions and φ a CTL- path formula.

If $TS \not\models \mathrm{A}\,\varphi$, then a counterexample for φ in TS can be determined in time $\mathcal{O}(N+M)$.

The same holds for a witness for φ, provided that $TS \models \mathrm{E}\,\varphi$.