Fairness in CTL

Lecture #22 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: `katoen@cs.rwth-aachen.de`

January 20, 2009

What did we treat so far?

- CTL semantics: for states, paths and transition systems
- CTL equivalence: e.g., expansion laws
- Existential normal form
- Expressivity of CTL versus LTL
- CTL model checking
- CTL*: extended CTL—expressivity and model checking

what about fairness in CTL?

Overview Lecture #22

$\Rightarrow$ Repetition: fairness in LTL

- Fair semantics for CTL
- CTL model checking with fairness
- Time complexity
- Summary of CTL model checking

Summary of action-based fairness

- *Fairness constraints* rule out unrealistic executions
  - by putting constraints on the actions that occur along infinite executions
- Unconditional, strong, and weak fairness constraints
  - unconditional $\Rightarrow$ strong fair $\Rightarrow$ weak fair
  - weak fairness rules out the least number of runs; unconditional the most
- *Fairness assumptions* allow distinct constraints on distinct action sets
- (Realizable) fairness assumptions are irrelevant for safety properties
  - important for the verification of liveness properties

LTL fairness constraints

Let Φ and Ψ be propositional logic formulas over *AP*.

1. An *unconditional LTL fairness constraint* is of the form:

$$ufair \;=\; \Box\Diamond\Psi$$

2. A *strong LTL fairness condition* is of the form:

$$sfair \;=\; \Box\Diamond\Phi \;\longrightarrow\; \Box\Diamond\Psi$$

3. A *weak LTL fairness constraint* is of the form:

$$wfair \;=\; \Diamond\Box\Phi \;\longrightarrow\; \Box\Diamond\Psi$$

Φ stands for "something is enabled"; Ψ for "something is taken"

LTL fairness assumption

- *LTL fairness assumption* = conjunction of LTL fairness constraints
  - the fairness constraints are of any arbitrary type
- Strong fairness assumption: $sfair \;=\; \bigwedge_{0<i\leqslant k} \left(\Box\Diamond\Phi_i \;\longrightarrow\; \Box\Diamond\Psi_i \right)$
- General format: $fair \;=\; ufair \;\wedge\; sfair \;\wedge\; wfair$
- Rules of thumb:
  - strong (or unconditional) fairness assumptions are useful for solving contentions
  - weak fairness suffices for resolving nondeterminism resulting from interleaving

Fair satisfaction

For state s in transition system *TS* (over *AP*) without terminal states, let

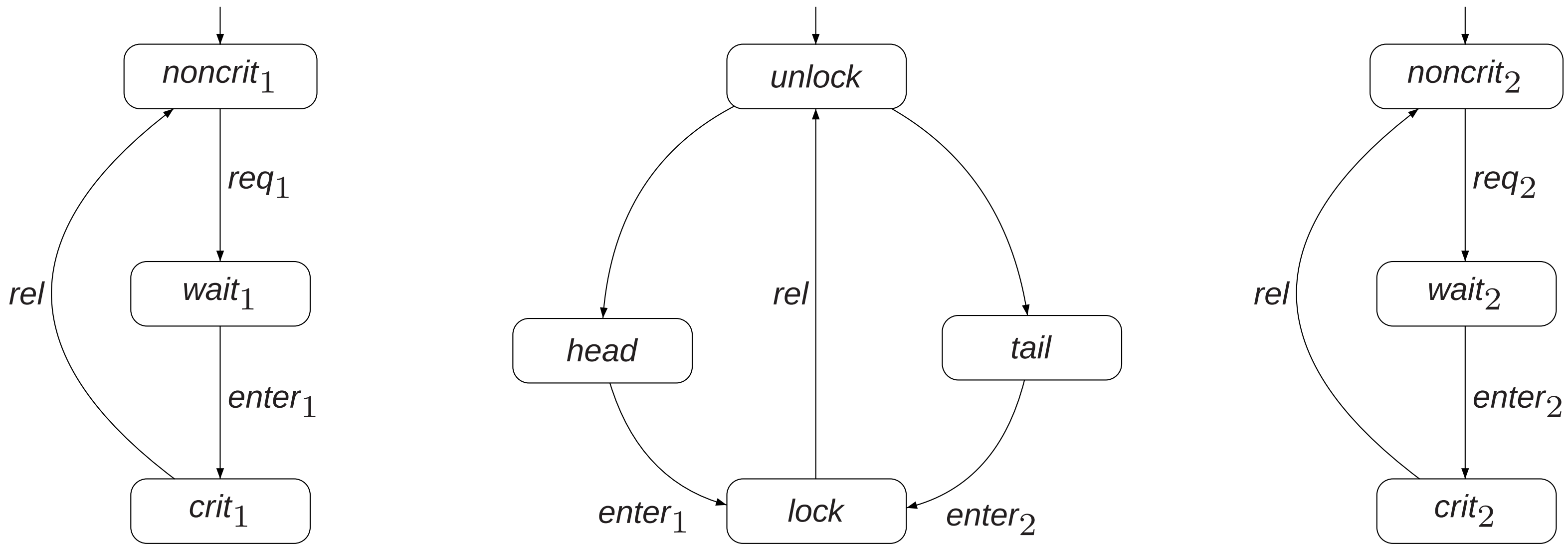
$$\mathit{FairPaths}_{fair}(s) = \{\, \pi \in \mathit{Paths}(s) \mid \pi \models \mathit{fair} \,\}$$
$$\mathit{FairTraces}_{fair}(s) = \{\, \mathit{trace}(\pi) \mid \pi \in \mathit{FairPaths}_{fair}(s) \,\}$$

For LTL-formula φ, and LTL fairness assumption $fair$:

$$s \models_{fair} \varphi \quad \text{if and only if} \quad \forall \pi \in \mathit{FairPaths}_{fair}(s).\, \pi \models \varphi \quad \text{and}$$
$$TS \models_{fair} \varphi \quad \text{if and only if} \quad \forall s_0 \in I.\, s_0 \models_{fair} \varphi$$

$\models_{fair}$ is the *fair satisfaction relation* for LTL; $\models$ the standard one for LTL

Randomized arbiter



$$TS_1 \parallel Arbiter \parallel TS_2 \not\models \Box\Diamond\, crit_1$$

But: $TS_1 \parallel Arbiter \parallel TS_2 \models_{fair} \Box\Diamond crit_1 \wedge \Box\Diamond crit_2$ with $fair = \Box\Diamond head \wedge \Box\Diamond tail$

Reducing $\models_{fair}$ to $\models$

For:

- transition system *TS* without terminal states
- LTL formula φ, and
- LTL fairness assumption *fair*

it holds:

$$\boxed{\textit{TS} \models_{fair} \varphi \qquad \text{if and only if} \qquad \textit{TS} \models (fair \rightarrow \varphi)}$$

verifying an LTL-formula under a fairness assumption can be done
using standard LTL model-checking algorithms

Overview Lecture #22

- Repetition: fairness in LTL

$\Rightarrow$ Fair semantics for CTL

- CTL model checking with fairness
- Time complexity
- Summary of CTL model checking

Fairness constraints in CTL

- For LTL it holds: *TS* $\models_{fair} \varphi$ if and only if *TS* $\models (fair \rightarrow \varphi)$
- An analogous approach for CTL is not possible!
- Formulas form $\forall(fair \rightarrow \varphi)$ and $\exists(fair \wedge \varphi)$ needed
- But: boolean combinations of path formulae are not allowed in CTL
- and: e.g., strong fairness constraints $\Box\Diamond b \rightarrow \Box\Diamond c \equiv \Diamond\Box\neg b \ \vee \ \Diamond\Box c$
  - cannot be expressed in CTL since persistence properties are not in CTL
- Solution: change the semantics of CTL by ignoring unfair paths

CTL fairness constraints

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

$$sfair = \bigwedge_{0<i\leqslant k} (\Box\Diamond\Phi_i \rightarrow \Box\Diamond\Psi_i)$$

  - where Φ_i and Ψ_i (for $0 < i \leqslant k$) are CTL-formulas over *AP*
  - weak and unconditional CTL fairness constraints are defined analogously, e.g.

$$ufair = \bigwedge_{0<i\leqslant k} \Box\Diamond\Psi_i \quad \text{and} \quad wfair = \bigwedge_{0<i\leqslant k} (\Diamond\Box\Phi_i \rightarrow \Box\Diamond\Psi_i)$$

  - a CTL fairness assumption $fair$ is a combination of $ufair$, $sfair$ and $wfair$

$\Rightarrow$ a CTL fairness constraint is an LTL formula over CTL state formulas!

  - note that $s \models \Phi_i$ and $s \models \Psi_i$ refer to standard (unfair!) CTL semantics

Semantics of fair CTL

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

$s \models_{fair} a$	iff	$a \in \mathit{Label}(s)$
$s \models_{fair} \neg\, \Phi$	iff	$\neg\, (s \models_{fair} \Phi)$
$s \models_{fair} \Phi \,\vee\, \Psi$	iff	$(s \models_{fair} \Phi) \,\vee\, (s \models_{fair} \Psi)$
$s \models_{fair} \exists\varphi$	iff	$\pi \models_{fair} \varphi$ for *some fair* path π that starts in s
$s \models_{fair} \forall\varphi$	iff	$\pi \models_{fair} \varphi$ for *all fair* paths π that start in s

$\pi \models_{fair} \bigcirc\Phi$	iff $\pi[1] \models_{fair} \Phi$
$\pi \models_{fair} \Phi \,\mathsf{U}\, \Psi$	iff $(\exists\, j \geqslant 0.\, \pi[j] \models_{fair} \Psi \;\wedge\; (\forall\, 0 \leqslant k < j.\, \pi[k] \models_{fair} \Phi))$

π is a fair path iff $\pi \models_{LTL} fair$ for CTL fairness assumption $fair$

Transition system semantics

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

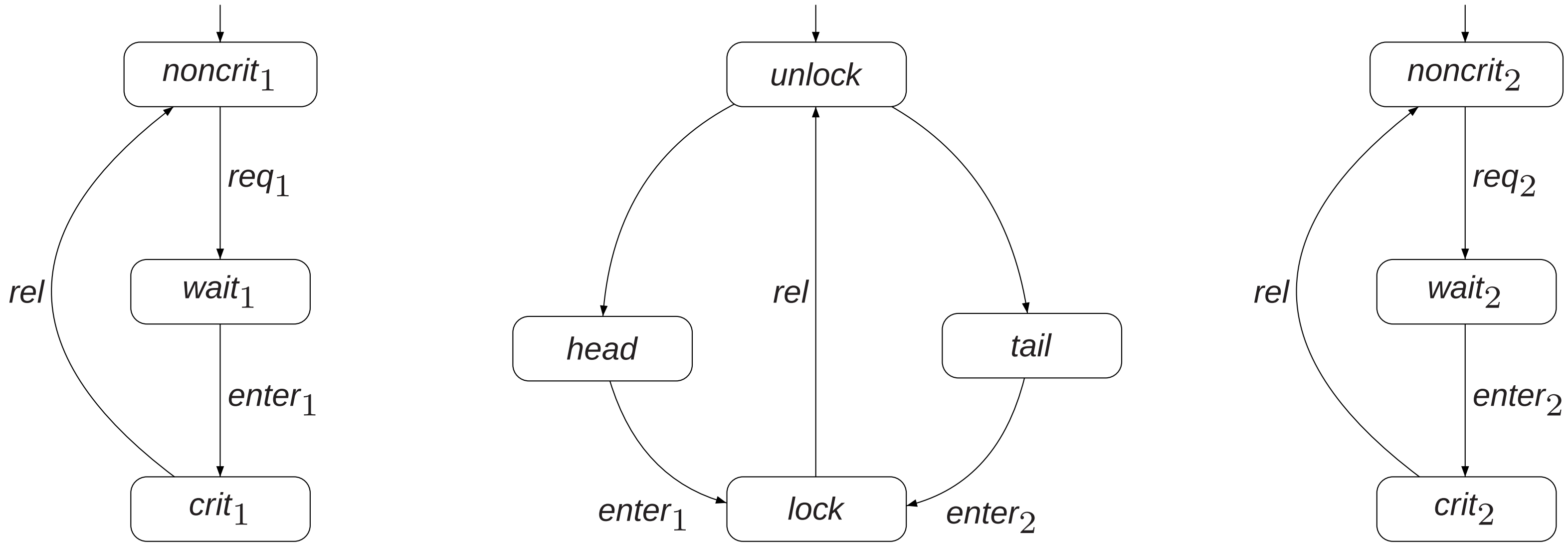
$$\mathit{Sat}_{fair}(\Phi) \; = \; \{ \, s \in S \mid s \models_{fair} \Phi \, \}$$

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

$$\mathit{TS} \models_{fair} \Phi \quad \text{if and only if} \quad \forall s_0 \in I. \, s_0 \models_{fair} \Phi$$

  - this is equivalent to $I \subseteq \mathit{Sat}_{fair}(\Phi)$

Randomized arbiter



$$TS_1 \parallel Arbiter \parallel TS_2 \not\models (\forall\Box\forall\Diamond\, crit_1) \;\wedge\; (\forall\Box\forall\Diamond\, crit_2)$$

But: $TS_1 \parallel Arbiter \parallel TS_2 \models_{fair} \forall\Box\forall\Diamond crit_1 \;\wedge\; \forall\Box\forall\Diamond crit_2$ with

$$fair \;=\; \Box\Diamond head \;\wedge\; \Box\Diamond tail$$

Overview Lecture #22

- Repetition: fairness in LTL
- Fair semantics for CTL

$\Rightarrow$ CTL model checking with fairness

- Time complexity
- Summary of CTL model checking

Fair CTL model-checking problem

For:

- finite transition system *TS* without terminal states
- CTL formula Φ in ENF, and
- CTL fairness assumption *fair*

establish whether or not:

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

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

CTL fairness constraints

- A *strong CTL fairness constraint*: $sfair = \bigwedge_{0<i\leqslant k} (\Box\Diamond\Phi_i \rightarrow \Box\Diamond\Psi_i)$
  - where Φ_i and Ψ_i (for $0 < i \leqslant k$) are CTL-formulas over *AP*

- Replace the CTL state-formulas in $sfair$ by fresh atomic propositions:

$$sfair := \bigwedge_{0<i\leqslant k} (\Box\Diamond a_i \rightarrow \Box\Diamond b_i)$$

  - where $a_i \in L(s)$ if and only if $s \in \mathit{Sat}(\Phi_i)$ (not $\mathit{Sat}_{fair}(\Phi_i)$!)
  - $\ldots$ $b_i \in L(s)$ if and only if $s \in \mathit{Sat}(\Psi_i)$ (not $\mathit{Sat}_{fair}(\Psi_i)$!)
  - (for unconditional and weak fairness this goes similarly)

- Note: $\pi \models fair$ iff $\pi[j..] \models fair$ for some $j \geqslant 0$ iff $\pi[j..] \models fair$ for all $j \geqslant 0$

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

$s \models_{fair} \exists\bigcirc a$ if and only if $\exists s' \in$ *Post*(s) with $s' \models a$ and *FairPaths*$(s') \neq \varnothing$

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

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in \textit{Paths}_{\textit{fin}}(s) \quad \text{with } n \geqslant 0$$

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

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

$s \models_{fair} \exists \bigcirc a$ if and only if $\exists s' \in \textit{Post}(s)$ with $s' \models a$ and $\underbrace{\textit{FairPaths}(s') \neq \varnothing}_{s' \models_{fair} \exists \Box \text{true}}$

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

$$s_0\, s_1\, s_2 \ldots s_{n-1} s_n \in \textit{Paths}_{\textit{fin}}(s) \quad \text{with } n \geqslant 0$$

such that $s_i \models a$ for $0 \leqslant i < n$, $s_n \models a'$, and $\underbrace{\textit{FairPaths}(s_n) \neq \varnothing}_{s_n \models_{fair} \exists \Box \text{true}}$

Basic algorithm

- Determine $\mathit{Sat}_{\mathit{fair}}(\exists\Box\text{true}) = \{ s \in S \mid \mathit{FairPaths}(s) \neq \varnothing \}$
- Introduce an atomic proposition a_{fair} and adjust labeling where:
  - $a_{\mathit{fair}} \in L(s)$ if and only if $s \in \mathit{Sat}_{\mathit{fair}}(\exists\Box\text{true})$
- Compute the sets $\mathit{Sat}_{\mathit{fair}}(\Psi)$ for all subformulas Ψ of Φ (in ENF) by:

$$
\begin{aligned}
\mathit{Sat}_{\mathit{fair}}(a) &= \{ s \in S \mid a \in L(s) \} \\
\mathit{Sat}_{\mathit{fair}}(\neg a) &= S \setminus \mathit{Sat}_{\mathit{fair}}(a) \\
\mathit{Sat}_{\mathit{fair}}(a \wedge a') &= \mathit{Sat}_{\mathit{fair}}(a) \cap \mathit{Sat}_{\mathit{fair}}(a') \\
\mathit{Sat}_{\mathit{fair}}(\exists\bigcirc a) &= \mathit{Sat}\,(\exists\bigcirc(a \wedge a_{\mathit{fair}})) \\
\mathit{Sat}_{\mathit{fair}}(\exists(a \,\mathsf{U}\, a')) &= \mathit{Sat}\,\big(\exists(a \,\mathsf{U}\, (a' \wedge a_{\mathit{fair}}))\big) \\
\mathit{Sat}_{\mathit{fair}}(\exists\Box a) &= \ldots\ldots
\end{aligned}
$$

- Thus: model checking CTL under fairness constraints is
  - CTL model checking + algorithm for computing $\mathit{Sat}_{\mathit{fair}}(\exists\Box a)$!

Model checking CTL with fairness

The model-checking problem for CTL with fairness can be reduced to:

- the model-checking problem for CTL (without fairness), and
- the problem of computing $\mathit{Sat}_{fair}(\exists\Box a)$ for $a \in AP$

note that $\exists\Box$true is a special case of $\exists\Box a$

thus a single algorithm suffices for $\mathit{Sat}_{fair}(\exists\Box a)$ and $\mathit{Sat}_{fair}(\exists\Box\text{true})$

Core model-checking algorithm

(* states are assumed to be labeled with a_i and b_i *)

compute *Sat*$_{fair}(\exists\Box\text{true}) \;=\; \{\, s \in S \mid$ *FairPaths*$(s) \neq \varnothing \,\}$
forall $s \in$ *Sat*$_{fair}(\exists\Box\text{true})$ **do** $L(s) := L(s) \cup \{\, a_{fair} \,\}$ **od**

(* compute *Sat*$_{fair}(\Phi)$ *)

for all $0 < i \leqslant |\,\Phi\,|$ **do**
  for all $\Psi \in$ *Sub*(Φ) with $|\,\Psi\,| = i$ **do**
    switch(Ψ):

true	:	*Sat*$_{fair}(\Psi) := S$;
a	:	*Sat*$_{fair}(\Psi) := \{\, s \in S \mid a \in L(s) \,\}$;
$a \;\wedge\; a'$	:	*Sat*$_{fair}(\Psi) := \{\, s \in S \mid a, a' \in L(s) \,\}$;
$\neg a$	:	*Sat*$_{fair}(\Psi) := \{\, s \in S \mid a \notin L(s) \,\}$;
$\exists\bigcirc a$	:	*Sat*$_{fair}(\Psi) :=$ *Sat*$(\exists\bigcirc(a \;\wedge\; a_{fair}))$;
$\exists(a \,\mathsf{U}\, a')$	:	*Sat*$_{fair}(\Psi) :=$ *Sat*$(\exists(a \,\mathsf{U}\, (a' \;\wedge\; a_{fair})))$;
$\exists\Box a$	:	compute *Sat*$_{fair}(\exists\Box a)$

    end switch
    replace all occurrences of Ψ (in Φ) by the fresh atomic proposition a_Ψ
    forall $s \in$ *Sat*$_{fair}(\Psi)$ **do** $L(s) := L(s) \cup \{\, a_\Psi \,\}$ **od**
  od
od
return $I \subseteq$ *Sat*$_{fair}(\Phi)$

Characterization of $Sat_{fair}(\exists\Box a)$

$$s \models_{sfair} \exists\Box a \quad \text{where} \quad sfair = \bigwedge_{0<i\leqslant k} (\Box\Diamond a_i \rightarrow \Box\Diamond b_i)$$

iff there exists a finite path fragment $s_0 \ldots s_n$ and a cycle $s'_0 \ldots s'_r$ with:

1. $s_0 = s$ and $s_n = s'_0 = s'_r$
2. $s_i \models a$, for any $0 \leqslant i \leqslant n$, and $s'_j \models a$, for any $0 \leqslant j \leqslant r$, and
3. *Sat*$(a_i) \cap \{ s'_1, \ldots, s'_r \} = \varnothing$ or *Sat*$(b_i) \cap \{ s'_1, \ldots, s'_r \} \neq \varnothing$ for $0 < i \leqslant k$

Proof

Computing $\mathit{Sat}_{fair}(\exists\Box a)$

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

  $\Rightarrow$ each infinite path fragment in $\mathit{TS}[a]$ satisfies $\Box a$

- $s \models_{fair} \exists\Box a$ iff there is a non-trivial SCC D in $\mathit{TS}[a]$ reachable from s:

  $D \cap \mathit{Sat}(a_i) = \varnothing$ or $D \cap \mathit{Sat}(b_i) \neq \varnothing$ for $0 < i \leqslant k$ (*)

- $\mathit{Sat}_{sfair}(\exists\Box a) = \{ s \in S \mid \mathit{Reach}_{\mathit{TS}[a]}(s) \cap T \neq \varnothing \}$
  - T is the union of all non-trivial SCCs C that contain D satisfying (*)

how to compute the set T of SCCs?

Unconditional fairness

$$ufair \equiv \bigwedge_{0<i\leqslant k} \Box\Diamond b_i$$

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

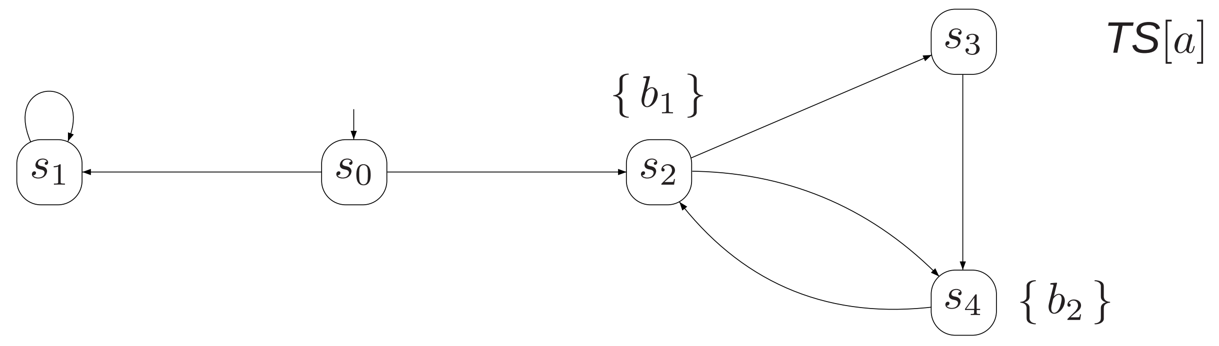
$$C \cap \textit{Sat}(b_i) \neq \varnothing \quad \text{for all } 0 < i \leqslant k$$

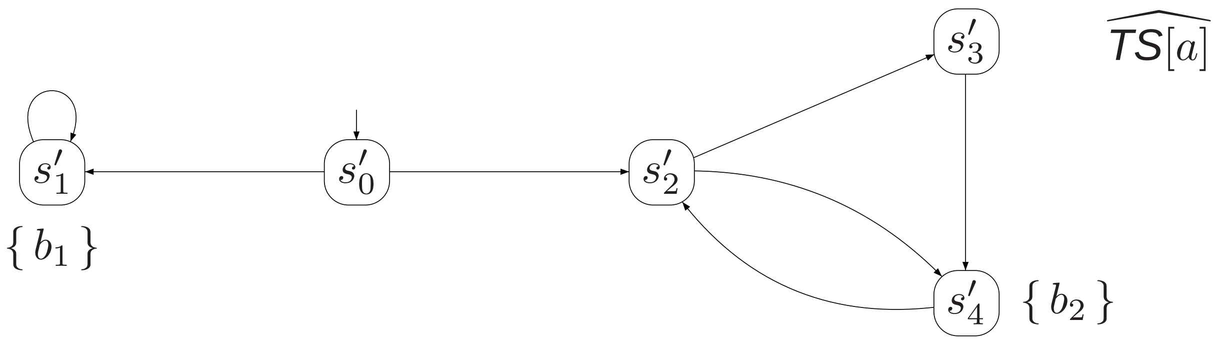
It now follows:

$$s \models_{ufair} \exists\Box a \quad \text{if and only if} \quad \textit{Reach}_{\textit{TS}[a]}(s) \cap T \neq \varnothing$$

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

Example





$TS[a] \models_{ufair} \exists\Box a$ but $\widehat{TS[a]} \not\models_{ufair} \exists\Box a$ with $ufair = \Box\Diamond b_1 \ \wedge \ \Box\Diamond b_2$

Strong fairness

- $sfair = \Box\Diamond a_1 \rightarrow \Box\Diamond b_1$, i.e., $k{=}1$

- $s \models_{sfair} \exists\Box a$ iff C is a non-trivial SCC in *TS*$[a]$ reachable from s with:

  (1) $C \cap$ *Sat*$(b_1) \neq \varnothing$, or

  (2) $D \cap$ *Sat*$(a_1) = \varnothing$, for some non-trivial SCC D in C

- D is a non-trivial SCC in the graph that is obtained from $C[\neg a_1]$

- For T the union of non-trivial SCCs in satisfying (1) and (2):

$$s \models_{sfair} \exists\Box a \quad \text{if and only if} \quad \mathit{Reach}_{\mathit{TS}[a]}(s) \cap T \neq \varnothing$$

for several strong fairness constraints ($k > 1$), this is applied recursively

T is determined by standard graph analysis (DFS)

Overview Lecture #22

- Repetition: fairness in LTL
- Fair semantics for CTL
- CTL model checking with fairness

$\Rightarrow$ Time complexity

- Summary of CTL model checking

Time complexity

For 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 $\mathcal{O}(|\,\Phi\,| \cdot (N + M) \cdot k)$

Overview Lecture #22

- Repetition: fairness in LTL
- Fair semantics for CTL
- CTL model checking with fairness
- Time complexity

$\Rightarrow$ Summary of CTL model checking

Summary of CTL model checking (1)

- CTL is a logic for formalizing properties over computation trees
- The expressiveness of LTL and CTL is incomparable
- Fairness constraints cannot be expressed in CTL
  - but are incorporated by adapting the CTL semantics such that quantification is over fair paths
- CTL model checking is by a recursive descent over parse tree of Φ
  - *Sat*$(\exists(\Phi \, \mathsf{U} \, \Psi))$ is determined using a least fixed point computation
  - *Sat*$(\exists\Box\Phi)$ is determined by a greatest fixed point computation

Summary of CTL model checking (2)

- Time complexity of CTL model-checking $\mathit{TS} \models \Phi$ is:
  - is linear in $|\mathit{TS}|$ and $|\Phi|$ and linear in k for k fairness constraints

- Checking $\mathit{TS} \models_{fair} \Phi$ is $\mathit{TS} \models \Phi$ plus computing $\mathit{Sat}_{fair}(\exists\Box a)$

- Counterexamples and witnesses for CTL path-formulae can be determined using graph algorithms

- CTL* is more expressive than both CTL and LTL

- The CTL* model-checking problem can be solved by an appropriate combination of the CTL and the LTL model-checking algorithm

- The CTL*-model checking problem is PSPACE-complete