Verification

Lecture 10

Bernd Finkbeiner
Peter Faymonville
Michael Gerke

UNIVERSITÄT DES SAARLANDES

REVIEW: CTL Syntax

modal logic over infinite trees [Clarke & Emerson 1981]

- State formulas
  - $a \in AP$ — atomic proposition
  - $\neg\Phi$ and $\Phi \wedge \Psi$ — negation and conjunction
  - E φ — there exists a path fulfilling φ
  - A φ — all paths fulfill φ
- Path formulas
  - X Φ — the next state fulfills Φ
  - Φ U Ψ — Φ holds until a Ψ-state is reached

$\Rightarrow$ note that X and U alternate with A and E

- AX X Φ and A EX $\Phi \notin$ CTL, but AX AX Φ and AX EX $\Phi \in$ CTL

Alternative syntax: E $\approx \exists$, A $\approx \forall$, X $\approx \bigcirc$, G $\approx \Box$, F $\approx \Diamond$.

REVIEW: Basic model checking algorithm

Require: finite transition system *TS* with states *S* and initial states *I*, and CTL formula Φ (both over *AP*)

Ensure: $TS \models \Phi$

{compute the sets $Sat(\Phi) = \{ q \in S \mid q \models \Phi \}$}

for all $i \leq |\Phi|$ **do**

  for all $\Psi \in Sub(\Phi)$ with $|\Psi| = i$ **do**

    compute $Sat(\Psi)$ from $Sat(\Psi')$ {for maximal proper $\Psi' \in Sub(\Psi)$}

  end for

end for

return $I \subseteq Sat(\Phi)$

REVIEW: Computing $Sat(\mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi))$ (3)

Require: finite transition system with states S CTL-formula $\mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi)$
Ensure: $Sat(\mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi)) = \{\, q \in S \mid q \models \mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi) \,\}$

$V := Sat(\Psi)$; {V administers states q with $q \models \mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi)$}
$T := V$; {T contains the already visited states q with $q \models \mathrm{E}\,(\Phi\,\mathrm{U}\,\Psi)$}
while $V \neq \varnothing$ **do**
  let $q' \in V$;
  $V := V \setminus \{\, q' \,\}$;
  for all $q \in Pre(q')$ **do**
    if $q \in Sat(\Phi) \setminus T$ **then** $V := V \cup \{\, q \,\}$; $T := T \cup \{\, q \,\}$; **endif**
  end for
end while
return T

REVIEW: Computing $Sat(\mathrm{EG}\,\Phi)$

$V := S \setminus Sat(\Phi)$; {$V$ contains any not visited q' with $q' \not\models \mathrm{E\,G}\,\Phi$}

$T := Sat(\Phi)$; {T contains any q for which $q \models \mathrm{E\,G}\,\Phi$ has not yet been disproven}

for all $q \in Sat(\Phi)$ **do** $c[q] := |Post(q)|$; **od** {initialize array c}

while $V \neq \varnothing$ **do**
  {loop invariant: $c[q] = |Post(q) \cap (T \cup V)|$}
  let $q' \in V$; {$q' \not\models \Phi$}
  $V := V \setminus \{ q' \}$; {$q'$ has been considered}
  for all $q \in Pre(q')$ **do**
    if $q \in T$ **then**
      $c[q] := c[q] - 1$; {update counter $c[q]$ for predecessor q of q'}
      if $c[q] = 0$ **then**
        $T := T \setminus \{ q \}$; $V := V \cup \{ q \}$; {q does not have any successor in T}
      end if
    end if
  end for
end while
return T

Time complexity

For transition system TS with N states and K edges,
and CTL formula Φ, the CTL model-checking problem $TS \models \Phi$
can be determined in time $\mathcal{O}(|\Phi| \cdot (N + M))$

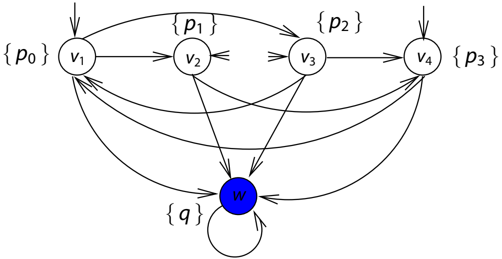
this applies to both algorithms for EG Φ

Model-checking LTL versus CTL

- Let *TS* be a transition system with *N* states and *M* edges
- Model-checking LTL-formula Φ has time-complexity $\mathcal{O}((N+M) \cdot 2^{|\Phi|})$
  - linear in the state space of the system model
  - exponential in the length of the formula
- Model-checking CTL-formula Φ has time-complexity $\mathcal{O}((N+M) \cdot |\Phi|)$
  - linear in the state space of the system model and the formula
- Is model-checking CTL more efficient?

Hamiltonian path problem

⇒ LTL-formulae can be exponentially shorter than their CTL-equivalent



- Existence of Hamiltonian path in LTL:
  $\bigwedge_i \big(\mathsf{F}\, p_i \;\wedge\; \mathsf{G}\,(p_i \rightarrow \mathsf{X}\,\mathsf{G}\,\neg p_i) \big)$
- In CTL, all possible (= 4!) routes need to be encoded

Equivalence of LTL and CTL formulas

CTL-formula Φ and LTL-formula φ (both over *AP*) are equivalent, denoted $\Phi \equiv \varphi$, if for any state graph *TS* (over *AP*):

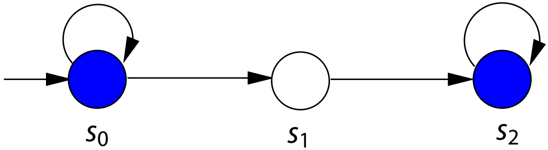
$$TS \models \Phi \quad \text{if and only if} \quad TS \models \varphi$$

Examples (1)

CTL-formula $\mathrm{A\,G\,A\,F}\,a$ and LTL-formula $\mathrm{G\,F}\,a$ are equivalent.

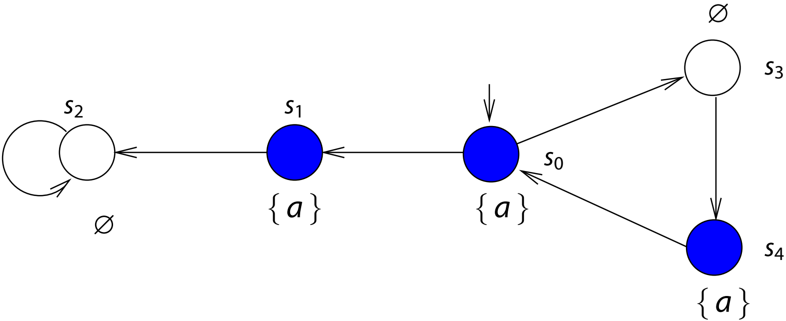
Examples (2)

AF AG a is not equivalent to F G a



Examples (3)

F ($a \wedge$ X a) is not equivalent to AF ($a \wedge$ AX a)



LTL and CTL are incomparable

- Some LTL-formulas cannot be expressed in CTL, e.g.,
  - $\mathrm{F}\,\mathrm{G}\,a$
  - $\mathrm{F}\,(a \wedge \mathrm{X}\,a)$
- Some CTL-formulas cannot be expressed in LTL, e.g.,
  - $\mathrm{AF}\,\mathrm{AG}\,a$
  - $\mathrm{AF}\,(a \wedge \mathrm{AX}\,a)$
  - $\mathrm{AG}\,\mathrm{EF}\,a$

$\Rightarrow$ Cannot be expressed = there does not exist an equivalent formula

Example

The CTL-formula AG EF a cannot be expressed in LTL

Comparing LTL and CTL

Let Φ be a CTL-formula, and φ the LTL-formula obtained by eliminating all path quantifiers in Φ. Then: [Clarke & Draghicescu]

$\Phi \equiv \varphi$ or there does not exist any LTL-formula that is equivalent to Φ

Comparing LTL and CTL

The LTL-formula FG *a* cannot be expressed in CTL

REVIEW: LTL Fairness constraints

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

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

$$ufair = \mathrm{G}\,\mathrm{F}\,\Psi$$

2. A strong LTL fairness condition (compassion) is of the form:

$$sfair = \mathrm{G}\,\mathrm{F}\,\Phi \longrightarrow \mathrm{G}\,\mathrm{F}\,\Psi$$

3. A weak LTL fairness constraint (justice) is of the form:

$$wfair = \mathrm{F}\,\mathrm{G}\,\Phi \longrightarrow \mathrm{G}\,\mathrm{F}\,\Psi$$

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

REVIEW: Fair satisfaction

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

$$FairPaths_{fair}(q) = \{ \pi \in Paths(q) \mid \pi \models fair \}$$

$$FairTraces_{fair}(q) = \{ trace(\pi) \mid \pi \in FairPaths_{fair}(q) \}$$

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

$$q \models_{fair} \varphi \quad \text{if and only if} \quad \forall \pi \in FairPaths_{fair}(q).\, \pi \models \varphi \quad \text{and}$$

$$TS \models_{fair} \varphi \quad \text{if and only if} \quad \forall q_0 \in Q_0.\, q_0 \models_{fair} \varphi$$

$\models_{fair}$ is the <u>fair satisfaction relation</u> for LTL; $\models$ the standard one for LTL

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

For:

- state graph *TS* without terminal states
- LTL formula φ, and
- LTL fairness assumption *fair*

it holds:

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

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

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 of form $\forall(fair \rightarrow \varphi)$ and $\exists(fair \wedge \varphi)$ needed
- But: boolean combinations of path formulas
  are not allowed in CTL
- and: strong fairness constraints

  $$\mathrm{G\,F}\, b \rightarrow \mathrm{G\,F}\, c \equiv \mathrm{F\,G}\, \neg b \vee \mathrm{F\,G}\, c$$

  cannot be expressed, 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:

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

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

$$ufair = \bigwedge_{0<i\leq k} \mathsf{G}\,\mathsf{F}\,\Psi_i \quad \text{and} \quad wfair = \bigwedge_{0<i\leq k} (\mathsf{F}\,\mathsf{G}\,\Phi_i \rightarrow \mathsf{G}\,\mathsf{F}\,\Psi_i)$$

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

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

Semantics of fair CTL

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

$s \models_{fair} a$	iff $a \in Label(s)$
$s \models_{fair} \neg \Phi$	iff $\neg (s \models_{fair} \Phi)$
$s \models_{fair} \Phi \vee \Psi$	iff $(s \models_{fair} \Phi) \vee (s \models_{fair} \Psi)$
$s \models_{fair} \mathrm{E}\, \varphi$	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:

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

- *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

- A strong CTL fairness constraint: $sfair = \bigwedge_{0<i\leq k} (\mathrm{G\,F}\,\Phi_i \rightarrow \mathrm{G\,F}\,\Psi_i)$
  - 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:

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

  - 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 $Sat_{fair}(\Psi_i)$!)
  - (for unconditional and weak fairness this goes similarly)
- Note: $\pi \models fair$ iff $\pi[j..] \models fair$ for some $j \geq 0$ iff $\pi[j..] \models fair$ for all $j \geq 0$

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

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

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

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

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

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

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

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

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

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

Basic algorithm

- Determine $Sat_{fair}(\text{EG true}) = \{ q \in Q \mid FairPaths(q) \neq \varnothing \}$
- 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:

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

- 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 $Sat_{fair}(\text{EG true}) = \{ q \in Q \mid FairPaths(q) \neq \varnothing \}$
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	:	$Sat_{fair}(\Psi) := Q$;
a	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a \in L(s) \}$;
$a \wedge a'$	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a, a' \in L(s) \}$;
$\neg a$	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a \notin L(s) \}$;
EX a	:	$Sat_{fair}(\Psi) := Sat(\text{EX}\,(a \wedge a_{fair}))$;
E $(a \,\text{U}\, a')$	:	$Sat_{fair}(\Psi) := Sat(\text{E}\,(a \,\text{U}\, (a' \wedge a_{fair})))$;
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 $0 < i \leq k$