FMSoft
Lecture 5 — Model checking for CTL

(pre-lecture version)

Tadeusz Litak

Nov 13, 2018

Informatik 8, FAU Erlangen-Nürnberg

- Why both smallest and greatest fixpoints are going to be important for us?
- Recall our goal: computing

$$[\![\phi]\!]^{\mathcal{M}} := \{s \in \mathcal{M} \mid s \vDash \phi\}$$

- There are some obvious functions $2^S \to 2^S$
- Consider $(\mathsf{EX})A := \{s \in S \mid \exists t.\, s \longrightarrow t \,\&\, t \in A\}$ ...
- ...and $(\mathsf{AX})A := \{s \in S \mid \forall t.\, s \longrightarrow t \Rightarrow t \in A\}$
- ...and $f_1(A) = [\![\phi]\!]^{\mathcal{M}} \cap (\mathsf{AX})A$ ...
- ...now contrast it with $f_2(A) = [\![\phi]\!]^{\mathcal{M}} \cup (\mathsf{EX})A$

Equivalences for fixpoint computation

- $\mathsf{AG}\phi \equiv \phi \wedge \mathsf{AX}\,\mathsf{AG}\phi$
- $\mathsf{EG}\phi \equiv \phi \wedge \mathsf{EX}\,\mathsf{EG}\phi$
- $\mathsf{AF}\phi \equiv \phi \vee \mathsf{AX}\,\mathsf{AF}\phi$
- $\mathsf{EF}\phi \equiv \phi \vee \mathsf{EX}\,\mathsf{EF}\phi$
- $\mathsf{A}[\phi \mathsf{U} \psi] \equiv \psi \vee (\phi \wedge \mathsf{AX}\,\mathsf{A}[\phi \mathsf{U} \psi])$
- $\mathsf{E}[\phi \mathsf{U} \psi] \equiv \psi \vee (\phi \wedge \mathsf{EX}\,\mathsf{E}[\phi \mathsf{U} \psi])$

Denotations as fixpoins

- $[\![\mathsf{AG}\phi]\!]^{\mathcal{M}} = [\![\phi]\!]^{\mathcal{M}} \cap (\mathsf{AX})[\![\mathsf{AG}\phi]\!]^{\mathcal{M}}$
- $[\![\mathsf{EG}\phi]\!]^{\mathcal{M}} = [\![\phi]\!]^{\mathcal{M}} \cap (\mathsf{EX})[\![\mathsf{EG}\phi]\!]^{\mathcal{M}}$
- $[\![\mathsf{AF}\phi]\!]^{\mathcal{M}} = [\![\phi]\!]^{\mathcal{M}} \cup (\mathsf{AX})[\![\mathsf{AF}\phi]\!]^{\mathcal{M}}$
- $[\![\mathsf{EF}\phi]\!]^{\mathcal{M}} = [\![\phi]\!]^{\mathcal{M}} \cup (\mathsf{EX})[\![\mathsf{EF}\phi]\!]^{\mathcal{M}}$
- $[\![\mathsf{A}[\phi \mathsf{U} \psi]]\!]^{\mathcal{M}} = [\![\psi]\!]^{\mathcal{M}} \cup ([\![\phi]\!]^{\mathcal{M}} \cap (\mathsf{AX})[\![\mathsf{A}[\phi \mathsf{U} \psi]]\!]^{\mathcal{M}})$
- $[\![\mathsf{E}[\phi \mathsf{U} \psi]]\!]^{\mathcal{M}} = [\![\psi]\!]^{\mathcal{M}} \cup ([\![\phi]\!]^{\mathcal{M}} \cap (\mathsf{EX})[\![\mathsf{E}[\phi \mathsf{U} \psi]]\!]^{\mathcal{M}})$

- Which one is greatest, which one is least?

- Now for actual model checking
- Recall: we could formulate CTL using EX, EU, AF as modal primitives

  and, say $\wedge$, $\neg$ as $\bot$ as propositional ones
- All of them computable using least fixpoints
- As it turns out, this is a suboptimal choice ...
- ...but let us describe this "pure least fixpoints" strategy first
- We compute $[\![\phi]\!]^{\mathcal{M}}$ passing through the model and labelling states with increasingly complex subformulas of ϕ

- nothing labelled with $\bot$
- clauses for $\psi_1 \wedge \psi_2$ and $\neg\psi$ obvious
- the clause for $\mathsf{EX}\psi$ obvious too
- $[\![\mathsf{AF}\psi]\!]^{\mathcal{M}} = [\![\psi]\!]^{\mathcal{M}} \cup (\mathsf{AX})[\![\mathsf{AF}\psi]\!]^{\mathcal{M}}$:
  - if a state labelled with ψ, label it with $\mathsf{AF}\psi$ ...
  - ...then the states whose all successors labelled with $\mathsf{AF}\psi$ ...
  - ...repeat the last step until no new states labelled

- $[\![\mathsf{E}[\psi_1 \mathsf{U} \psi_2]]\!]^{\mathcal{M}} = [\![\psi_2]\!]^{\mathcal{M}} \cup ([\![\psi_1]\!]^{\mathcal{M}} \cap (\mathsf{EX})[\![\mathsf{E}[\psi_1 \mathsf{U} \psi_2]]\!]^{\mathcal{M}})$
  - if a state labelled with ψ_2, label it with $\mathsf{E}[\psi_1 \mathsf{U} \psi_2]$ ...
  - ...then label these states which are already labelled with ψ_1 and have a successor with $\mathsf{E}[\psi_1 \mathsf{U} \psi_2]$ ...
  - ...repeat the last step until no new states labelled
- Note on complexity: a naïve implementation yields $O(|\phi| * |S|^2 * |\longrightarrow|)$

  With some care in the implementation of AF labelling, we should be able to get down to $O(|\phi| * |S| * (|S| + |\longrightarrow|))$ claimed by Huth&Ryan *Logic in Computer Science* book, § 3.6 on model-checking algorithms
- linear in the size of the formula, quadratic in the size of the model

Improvement

- make sure you do, e.g., backwards breadth-first search to avoid visiting same mode
- replace AF with EG
- $[\![\mathsf{EG}\psi]\!]^{\mathcal{M}} = [\![\psi]\!]^{\mathcal{M}} \cap (\mathsf{EX})[\![\mathsf{EG}\psi]\!]^{\mathcal{M}}$
- This needs greatest fixpoint!
- ...you need to start with ψ and keep deleting points ...
- ...find the maximal *strongly connected components (SCCs)* among those satisfying ψ ...

  This is so-called *Tarjan's algorithm*

  Does not revisit nodes, forms a spanning forest of search trees, SCCs recovered as its subtrees
- is it enough?

- ...no, one needs to find (backwards breadth-first search?) all points from which a ψ-SCC is reachable
- Complexity $O(|\phi| * (|S| + | \longrightarrow |))$

  Huth&Ryan, § 3.6

  Baier & Katoen, *Principles of Model Checking*, § 6.4.3
- ...linear both in the size of the formula and the size of the model!

Fairness?

- We mentioned liveness and especially fairness
- recall FAIRNESS keyword in NuSMV/nuXmv ...
- how would they fare here?

- strong fairness condition: a conjunction of the form

$$\bigwedge_{i \leq n} (\mathsf{GF}\phi_i \to \mathsf{GF}\psi_i)$$

  where ϕ_i, ψ_i are CTL formulas
- weak fairness condition: a conjunction of the form

$$\bigwedge_{i \leq n} (\mathsf{FG}\phi_i \to \mathsf{GF}\psi_i)$$

  where ϕ_i, ψ_i are CTL formulas
- unconditional fairness condition: a conjunction of the form

$$\bigwedge_{i \leq n} \mathsf{GF}\psi_i$$

  where ψ_i's are CTL formulas
- More generally, a fairness condition C is any conjunction of the above three

  Baier and Katoen, *Principles of Model Checking*, § 6.5

- Consider a path π in $\mathcal{M}$, ϕ, ψ CTL-formulas
- $\mathcal{M}, \pi \vDash \mathsf{GF}\phi$ if for infinitely many i, $\mathcal{M}, \pi(i) \vDash \phi$
- $\mathcal{M}, \pi \vDash \mathsf{GF}\phi \to \mathsf{GF}\psi$ if ...
- ...whenever ϕ holds on infinitely many points, so does ψ
- $\mathcal{M}, \pi \vDash \mathsf{FG}\phi \to \mathsf{GF}\psi$ if ...
- ...whenever ϕ holds on some suffix of π, ψ holds on infinitely many points
- A path is C-fair if it satisfies all the conjuncts of C
- $\Pi_C(s)$ is the set of all C-fair paths starting at s

Presenting FCTL

- Now let us extend the language of CTL with
- $\ldots \mathsf{E}_C\mathsf{X}\phi \mid \mathsf{E}_C[\phi \mathsf{U} \psi] \mid \mathsf{A}_C[\phi \mathsf{U} \psi]$
- $\mathcal{M}, s \vDash \mathsf{E}_C\mathsf{X}\phi$ if exists $\pi \in \Pi_C(s)$ s.t. $\mathcal{M}, \pi(1) \vDash \phi$
- $\mathcal{M}, s \vDash \mathsf{E}_C[\phi \mathsf{U} \psi]$ if exists $\pi \in \Pi_C(s)$ and $n \in \mathbb{N}$ s.t. $\mathcal{M}, \pi(n) \vDash \psi$ and for any $i < n$, $\mathcal{M}, \pi(i) \vDash \phi$
- $\mathcal{M}, s \vDash \mathsf{A}_C[\phi \mathsf{U} \psi]$ if for all $\pi \in \Pi_C(s)$ there exists $n \in \mathbb{N}$ s.t. $\mathcal{M}, \pi(n) \vDash \psi$ and for any $i < n$, $\mathcal{M}, \pi(i) \vDash \phi$
- Other connectives work in a similar way

Any improvements in expressivity?

- Consider even the simplest unconditional fairness condition $\mathsf{GF}\neg\mathtt{idle}$
- How would you express $\mathsf{A}_{\mathsf{GF}\neg\mathtt{idle}}\mathsf{G}\phi$ in ordinary CTL?
- In LTL you simply write $\mathsf{GF}\neg\mathtt{idle} \rightarrow \mathsf{G}\phi$
- $\mathsf{AG}(\mathsf{AGAF}\neg\mathtt{idle} \rightarrow \phi)$ does not have the same meaning ...

Extending the model checking algorithm

- As before, $\mathsf{E}_C\mathsf{G}$, $\mathsf{E}_C\mathsf{X}$ and $\mathsf{E}_C\mathsf{U}$ form a sufficient set of connectives
- Moreover, we have additional equivalences:

$$\mathsf{E}_C[\phi \mathsf{U} \psi] \equiv \mathsf{E}[\phi \mathsf{U} (\psi \wedge \mathsf{E}_C\mathsf{G}\top)]$$
$$\mathsf{E}_C\mathsf{X}\phi \equiv \mathsf{EX}(\phi \wedge \mathsf{E}_C\mathsf{G}\top)$$

- Proof sketch: π satisfies C iff all its suffixes do
  In other words, a single finite prefix is irrelevant anyway
- Thus, we just need to extend the algorithm with $\mathsf{E}_C\mathsf{G}\phi$
- We also need to pre-compute extensions of all CTL subformulas used in C

Improvement

- make sure you do, e.g., backwards breadth-first search to avoid visiting same mode
- for $[\![\mathsf{E}_C\mathsf{G}\psi]\!]^{\mathcal{M}}$, you again start with deleting non-ψ points ...
- ...find the maximal *strongly connected components (SCCs)* among those satisfying ψ ...

  our old friend Tarjan's algorithm
- and furthermore depending on C ...

- $\mathcal{M}, \pi \models \mathsf{GF}\phi$ if for infinitely many i, $\mathcal{M}, \pi(i) \models \phi$
- ...to check $\mathsf{E}_{\mathsf{GF}\phi}\mathsf{G}\psi$, delete all SCC's with no ϕ
- $\mathcal{M}, \pi \models \mathsf{GF}\phi \to \mathsf{GF}\chi$ if it is not the case that ϕ holds on infinitely many points and yet χ holds only on finitely many points
- ...to check $\mathsf{E}_{\mathsf{GF}\phi\to\mathsf{GF}\chi}\mathsf{G}\psi$, delete all SCC's where ϕ occurs, but χ does not
- $\mathcal{M}, \pi \models \mathsf{FG}\phi \to \mathsf{GF}\chi$ if it is not the case that ϕ holds on some suffix of π and yet χ only on finitely many points
- ...to check $\mathsf{E}_{\mathsf{FG}\phi\to\mathsf{GF}\chi}\mathsf{G}\psi$, delete all SCC's where ϕ holds everywhere, but ψ nowhere

  remember that a SCC represents a suffix rather than the entire path

Finish as before

- one needs to find (backwards breadth-first search?) all points from which a ψ-SCC is reachable
- same procedure for every subformula of ϕ of the form $\mathsf{E}_C\mathsf{G}\psi$
- Complexity $O(|\phi| * |C| * (|S| + |\longrightarrow|)$
- ...still linear both in $|\phi|$ and in $|S|$!
- Of course, rather awkward syntax and semantics

Aside on CTL^f

- A new extension CTL^f (fair CTL) proposed by Ghilardi and van Gool at LiCS 2016

  Deeper mathematical motivation, no actual model checking in that paper
- Instead of these fairness constraints and all the new connectives like $\mathsf{E}_C\mathsf{G}$, $\mathsf{E}_C\mathsf{X}$ and $\mathsf{E}_C\mathsf{U}$ ...
- ...just return to ordinary CTL and replace EG with $\mathsf{E}[\phi\mathsf{G}\psi]$
- $\mathcal{M}, s \vDash \mathsf{E}[\phi\mathsf{G}\psi]$ if for some $\pi \in \Pi(s)$, ϕ holds at all points of π and ψ holds at infinitely many points of π
- The old $\mathsf{EG}\phi$ is expressible as $\mathsf{E}[\phi\mathsf{G}\top]$

- Binary EG expresses directly unconditional fairness ...
- ...$\mathsf{E}_{\mathsf{GF}\phi}\mathsf{G}\psi$ is $\mathsf{E}[\psi \mathsf{G} \phi]$
- How do you express weak fairness $\mathsf{E}_{\mathsf{FG}\phi \rightarrow \mathsf{GF}\chi}\mathsf{G}\psi$?
- Note that $(\mathsf{FG}\phi \rightarrow \mathsf{GF}\chi) \equiv_{\mathsf{LTL}} (\mathsf{GF}\neg\phi \vee \mathsf{GF}\chi)$
- Hence, you can do it as $\mathsf{E}[\psi \mathsf{G} \neg\phi] \vee \mathsf{E}[\psi \mathsf{G} \chi]$
- Can you express strong fairness $\mathsf{E}_{\mathsf{GF}\phi \rightarrow \mathsf{GF}\chi}\mathsf{G}\psi$?
- ...check it out!

- However, in order to compare LTL and CTL systematically, let us consider something still more powerful
- CTL* (Emerson and Clarke 1986), a language whose syntax incorporates both
  - explicit path formulas and
  - explicit state formulas
- Price: model checking no longer polynomial in $|\psi|$
- In fact, it can be done by reduction to model checking for LTL that Christoph is going to discuss
- Still more powerful: fixpoint calculi and parity games

  Beyond the scope of this lecture but amazingly effective