FMSoft
Lecture 3 — Introducing CTL

Tadeusz Litak

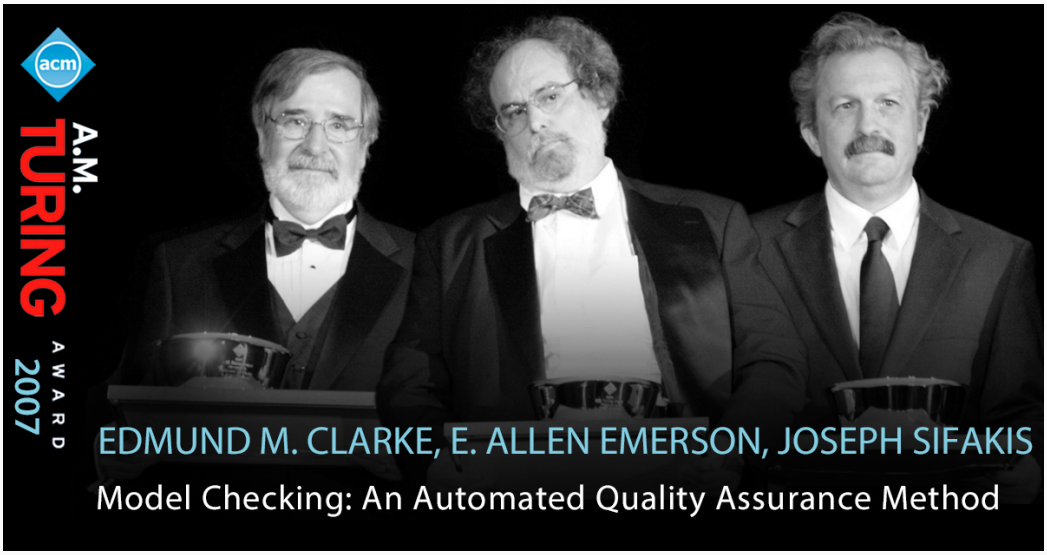
October 30, 2018 (pre-lecture version)

Informatik 8, FAU Erlangen-Nürnberg

Reminder of LTL problems

- Model-checking is not entirely trivial (in $|\phi|$)
- Expressing properties requiring combinations of universal and existential quantifiers over paths, such as reachability

- Almost simultaneously with Pnueli, another paradigm took off ...and a long sectarian war began
- CTL: Computation Tree Logic
- Clarke and Emerson in 1981, a similar formalism Queille and Sifakis in 1982



Compare citations ...

- Pnuelli: 1996 Turing Award
- Citation: *For seminal work introducing temporal logic into computing science and for outstanding contributions to program and systems verification*
- Clarke, Emerson, Sifakis: 2007 Turing Award
- Citation: *For their roles in developing model checking into a highly effective verification technology, widely adopted in the hardware and software industries*

- In CTL, one uses LTL modalities to build state formulas …
- …by prefixing these modalities with path quantifiers:
- A … *for all paths starting in this state*
  AX, AF, AG, AU
- E … *for some path starting in this state*
  EX, EF, EG, EU

- Model-checking works beautifully:
- linear both in $|M|$ and in $|\phi|$!
- Some LTL-expressible properties not expressible anymore though
- From the point of expressivity, no inclusion in either direction

  We'll return to it later

- Most painfully, we lose fairness properties
- But model-checking algorithm(s) can be easily extended to accommodate fairness constraints
- Furthermore, extensions proposed to explicitly include fairness
- Emerson and Lei 1986: FCTL, i.e., CTL with fairness constraints
- Ghilardi and Van Gool 2016: CTL^f, i.e., fair CTL
- Again, much more to come later

Defining CTL formally

- $\phi, \psi ::= \top \mid \mathsf{p} \mid \neg\phi \mid \phi \wedge \psi \mid \mathsf{EX}\phi \mid \mathsf{E}[\phi \mathsf{U} \psi] \mid \mathsf{A}[\phi \mathsf{U} \psi]$
- Semantics defined directly in terms of states!
- $\mathcal{M}, s \vDash \top$ always
- $\mathcal{M}, s \vDash \mathsf{p}$ if $\mathsf{p} \in L(s)$
- $\mathcal{M}, s \vDash \neg\phi$ if $\mathcal{M}, s \nvDash \phi$
- $\mathcal{M}, s \vDash \phi \wedge \psi$ if $\mathcal{M}, s \vDash \phi$ and $\mathcal{M}, s \vDash \psi$
- $\mathcal{M}, s \vDash \mathsf{EX}\phi$ if exists $\pi \in \Pi(s)$ s.t. $\mathcal{M}, \pi(1) \vDash \phi$

  What does it mean?
- $\mathcal{M}, s \vDash \mathsf{E}[\phi \mathsf{U} \psi]$ if exists $\pi \in \Pi(s)$ and $n \in \mathbb{N}$ s.t. $\mathcal{M}, \pi(n) \vDash \psi$ and for any $i < n$, $\mathcal{M}, \pi(i) \vDash \phi$
- $\mathcal{M}, s \vDash \mathsf{A}[\phi \mathsf{U} \psi]$ if for all $\pi \in \Pi(s)$ there exists $n \in \mathbb{N}$ s.t. $\mathcal{M}, \pi(n) \vDash \psi$ and for any $i < n$, $\mathcal{M}, \pi(i) \vDash \phi$

CTL **modalities**

- Defined abbreviations: usual booleans plus

$$\mathsf{AX}\phi := \neg\mathsf{EX}\neg\phi$$

- $$\mathsf{AF}\phi := \mathsf{A}[\top\mathsf{U}\phi] \quad \mathsf{EG}\phi := \neg\mathsf{AF}(\neg\phi)$$
  $$\mathsf{EF}\phi := \mathsf{E}[\top\mathsf{U}\phi] \quad \mathsf{AG}\phi := \neg\mathsf{EF}(\neg\phi)$$
- Exercise: Work out satisfaction clauses for the defined connectives!

Equivalences (of state formulas)

- We say that ϕ and ψ are (semantically) equivalent, notation $\phi \equiv \psi$, if for all $\mathcal{M}$ and for all states s in $\mathcal{M}$

$$\mathcal{M}, s \vDash \phi \text{ iff } \mathcal{M}, s \vDash \psi$$

- Note this can be stated in terms of defined bi-implication (equivalence connective)
- $\phi \equiv \psi$ if for all $\mathcal{M}$ and for all states s in $\mathcal{M}$,

$$\mathcal{M}, s \vDash \phi \leftrightarrow \psi$$

Exercise: work out the definitions of release and weak until!

Equivalences for fixpoint computation

- $\mathsf{AG}\phi \equiv \phi \wedge \mathsf{AXAG}\phi$
- $\mathsf{EG}\phi \equiv \phi \wedge \mathsf{EXEG}\phi$
- $\mathsf{AF}\phi \equiv \phi \vee \mathsf{AXAF}\phi$
- $\mathsf{EF}\phi \equiv \phi \vee \mathsf{EXEF}\phi$
- $\mathsf{A}[\phi \mathsf{U} \psi] \equiv \psi \vee (\phi \wedge \mathsf{AXA}[\phi \mathsf{U} \psi])$
- $\mathsf{E}[\phi \mathsf{U} \psi] \equiv \psi \vee (\phi \wedge \mathsf{EXE}[\phi \mathsf{U} \psi])$

Recursive equations?

- Fix two (we never need more) *recursive variables* x, y
- $\phi, \psi ::= x \mid y \mid \top \mid \bot \mid \mathsf{p} \mid \neg\mathsf{p} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \mathsf{EX}\phi \mid \mathsf{AX}\phi$
- We want solve "equations" of the form

$$x \text{ "=" } \phi x \ldots$$

- or maybe even

$$\mathit{Connective}(x, y) \text{ "=" } \phi(x, y, \mathit{Connective}(x, y)).$$

- Is it clear what such equations could mean?
- Do such equations always have a solution?
- Is such a solution always unique?
- Recall our goal: computing

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

- Actually, our next step is to focus on the fixpoint business
- ...and use it as a route to model-checking algorithms
- (it will also return to haunt us much later with denotational semantics of programs)
- In order to do things cleanly, some abstract spadework is needed