

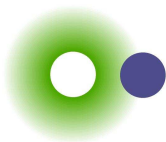
COMP6463: Temporal Logic and Model Checking

2. CTL

Michael Norrish

Canberra Research Lab., NICTA

Semester 2, 2010



NICTA

Outline

① Introduction

② Syntax

③ Semantics

④ Towards Model Checking

Temporal Logic

- 1950s: Arthur Prior's **tense logic** (before Kripke)
- 1970s: Temporal logic applied to programs (Burstall, Pnueli, Lamport)
- 1980s: **Model checking** (Emerson, Clarke and others)
 - Hardware examples studied
 - **Symbolic** Model Checking (thanks to BDDs)
- 1990s: Model checking in (hardware) industry
 - Symbolic Trajectory Evaluation in use at Intel
- 2000s: Industrial standardisation (IEEE) of PSL language

References

Model Checking, by Edmund Clarke, Orna Grumberg and Doron Peled. MIT Press, 1999.

Lots of stuff on the web.

Past and Future?

Last lecture's example logic featured past and future modalities (P and F).

- The logic was **bi-modal**

Logics in remaining lectures will be forward-looking only.

This is driven by computer science applications:

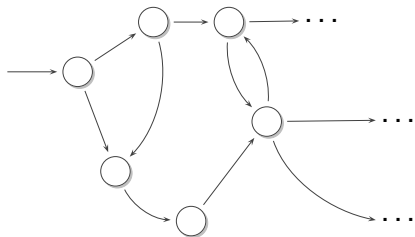
- We describe systems that have **starting states** . . .
- . . . and worry about what they may do subsequently.

Branching Time

We want to describe **non-deterministic** systems.

- Sometimes the non-determinism just comes from different behaviours in response to inputs.
- The inputs are uncontrolled so the system can behave “non-deterministically”.
- This is known as **input non-determinism**.

A system may also just be intrinsically non-deterministic:



Branches Lead to CTL

CTL stands for **Computation Tree Logic**.

Branches Lead to Two Sorts of Quantifier

In the universal case, we want to be able to say:

- 1 Something happens on every path originating from a world
 - Whichever path is taken from world w , p holds at some point
- 2 Something happens all the way along a particular path
 - On every state in the path, $p \Rightarrow q$ is true.

Existential analogues should be clear.

We will need to formalise the notion of **path**

Paths

An R -path π is an infinite sequence of states/worlds (in other words, $\pi : \mathbb{N} \rightarrow W$)

$$\pi_0, \pi_1, \pi_2, \pi_3 \dots$$

such that $\forall i. R(\pi_i, \pi_{i+1})$.

Define

$$R\text{-path}(\pi, w) \equiv \pi_0 = w \wedge \forall i. R(\pi_i, \pi_{i+1})$$

Read: “ π is an R -path starting at w ”

Discrete Time

Computer systems are usually thought of in terms of discrete time.

It makes sense to talk of “the next state” or “the next state along a particular path”.

In the hardware world, this is often enforced by the use of a clock.

Contrast **dense time**, which can be captured by the axiom

$$\diamond p \Rightarrow \diamond \diamond p$$

“If I can take one step to get to a state satisfying p , then I can take two steps to get to a state satisfying p ”

Basic CTL Syntax

$$\begin{aligned} \phi &::= p, q, r \dots \\ &| \neg \phi \\ &| \phi_1 \Rightarrow \phi_2 \\ &| AX \phi \\ &| EX \phi \\ &| A[\phi_1 \cup \phi_2] \\ &| E[\phi_1 \cup \phi_2] \end{aligned}$$

Define additional propositional connectives \wedge , \vee , \perp according to taste.

CTL's Modalities

The A in AX and $A[_ U _]$ stands for “All paths”.

$AX \phi$ (ne**X**t) “In every successor state, ϕ is true.”

$A[\phi_1 U \phi_2]$ (**U**ntil) “In all paths starting here, there is a state where ϕ_2 is true; until that point ϕ_1 is true.”

Existential Path Modalities

Existential path modalities have E instead of A : “there is a path”.

EX can be derived from AX :

$$EX \phi \equiv \neg AX \neg \phi$$

“It is not the case that every successor has $\neg \phi$ true”

“There exists a successor where ϕ is true”

Existential Path Modalities—Exists Until

$E[\phi_1 \cup \phi_2]$: there exists a path starting here, where ϕ_2 is eventually true, and at all worlds leading up to that point, ϕ_1 is true

Unfortunately $E[\phi_1 \cup \phi_2]$ is not simply the dual of $A[\phi_1 \cup \phi_2]$

- Both “until” modalities have an existential component: there exists a state on the path(s) where ϕ_2 holds

Derived Modalities

Substitute \top for ϕ_1 in the until modalities:

$$AF \phi \equiv A[\top \cup \phi]$$

$$EF \phi \equiv E[\top \cup \phi]$$

$AF \phi$: on all paths, there eventually exists a world where ϕ holds (before that point, “true” holds of the state)

$EF \phi$: there exists a path where ϕ eventually holds

We saw an F operator in the last lecture: it is the **future** operator.

More Derived Modalities

Take the duals of AF and EF :

$$AG \phi \equiv \neg EF \neg \phi$$

$$EG \phi \equiv \neg AF \neg \phi$$

$AG \phi$: there does not exist a path where $\neg \phi$ is eventually true. I.e., on all paths, $\neg \phi$ never holds. I.e., on all paths, ϕ always holds.

$EG \phi$: there exists a path where ϕ always holds

The G operator is the “going to be true”, or “globally” operator.

Notes on Until

Until is a complicated operator.

Read $_{-}[\phi_1 \cup \phi_2]$ as “ ϕ_1 is true until ϕ_2 holds”.

Depending on your sense of English, you might think that ϕ_2 is permitted not to hold. **This is NOT the case!**

Notes on Until

Until is a complicated operator.

Read $\neg[\phi_1 \text{ U } \phi_2]$ as “ ϕ_1 is true until ϕ_2 holds”.

Depending on your sense of English, you might think that ϕ_2 is permitted not to hold. **This is NOT the case!**

Some people define W “waiting for” to have that sense:

$\neg[\phi_1 W \phi_2]$: ϕ_2 may or may not come to be true on the path; until it does ϕ_1 holds.

(Clarke *et al* have $\neg[\phi_2 R \phi_1]$ (“releases”) instead.)

Exercise

Give the semantics of $A[\phi_1 W \phi_2]$

- 1 by defining it in terms of the existing modal operators
- 2 directly, by giving a condition on paths and worlds (as per the definitions later in this lecture)

Prove that the two definitions are equivalent.

Nailing the Semantics

You should already have an idea of what the operators mean.

But you may not be able to answer questions like:

- if $AG \phi$ is true at w , is ϕ true at w ?
- Is $A[\perp \cup \phi] \equiv \perp$?
- Is $A[\phi \cup \perp] \equiv \perp$?

CTL Semantics—Introduction

Let \mathfrak{M} be a triple of

- W a set of worlds/states
- R the accessibility relation between worlds
- V the valuation of propositional variables in the worlds

Have the usual $\mathfrak{M}, w \models \phi$ meaning

“ ϕ is true at world w in model \mathfrak{M} ”

CTL Semantics—Basic Operators

Write $\mathfrak{M}, w \models \phi$ to mean “ ϕ is true at world w in model \mathfrak{M} ”.

$\mathfrak{M}, w \models p$ if $p \in V(w)$

$\mathfrak{M}, w \models \neg\phi$ if $\mathfrak{M}, w \not\models \phi$

$\mathfrak{M}, w \models \phi_1 \Rightarrow \phi_2$ if $\mathfrak{M}, w \not\models \phi_1$ or $\mathfrak{M}, w \models \phi_2$

$\mathfrak{M}, w \models AX \phi$ if at every v such that $w R v$, have $\mathfrak{M}, v \models \phi$

$\mathfrak{M}, w \models EX \phi$ if there exists a v such that $w R v$ and $\mathfrak{M}, v \models \phi$

CTL Semantics—Until Modalities

Write $\mathfrak{M}, w \models \phi$ to mean “ ϕ is true at world w in model \mathfrak{M} ”.

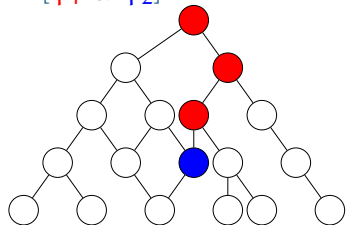
$\mathfrak{M}, w \models A[\phi_1 \cup \phi_2]$ if for every π such that $\text{R-path}(\pi, w)$, there exists an $i \geq 0$ such that $\mathfrak{M}, \pi(i) \models \phi_2$ and for all $0 \leq j < i$, $\mathfrak{M}, \pi(j) \models \phi_1$

$\mathfrak{M}, w \models E[\phi_1 \cup \phi_2]$ if there exists a π such that $\text{R-path}(\pi, w)$ and there also exists an $i \geq 0$ such that $\mathfrak{M}, \pi(i) \models \phi_2$ and for all $0 \leq j < i$, $\mathfrak{M}, \pi(j) \models \phi_1$

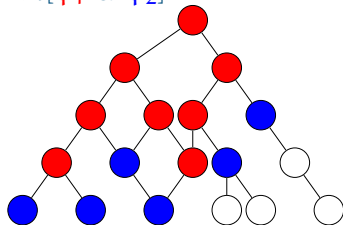
Note that the semantics breaks down a little if there isn't always at least one path out of a given world. (R should be **total**; W can still be finite.)

Graphically

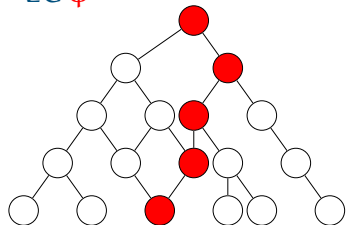
$E[\phi_1 \cup \phi_2]$



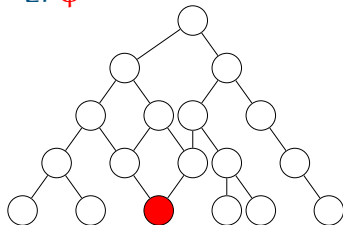
$A[\phi_1 \cup \phi_2]$



$EG \phi$



$EF \phi$



Proof Theory?

Bald Assertion: No-one cares about the problem of validity of CTL formulas over all frames

Contrast modal logic research into K, S4, S5, etc.

Instead, the interest is in verifying CTL properties of particular systems.

Is property ϕ true of system M ?

This is the **model checking** problem.

Some Typical Model Checking Assertions

$EF(Start \wedge \neg Ready)$: it is possible to get to a state where *Start* holds but *Ready* does not hold.

$AG(Req \Rightarrow AF Ack)$: if a request occurs, then it will be acknowledged eventually.

$AG(AF DeviceEnabled)$: The variable *DeviceEnabled* holds infinitely often on every computation path

$AG(EF Restart)$: From any state, it is possible to get to a *Restart* state.

A Minimal Expression of the Logic

Use only existential modalities. I.e.:

- propositional variables
- \neg and \vee
- EX
- EG
- $E[_ \cup _]$

AX from EX is easy (dualise)

$$A[\phi_1 \cup \phi_2] \equiv \neg E[\neg\phi_2 \cup (\neg\phi_1 \wedge \neg\phi_2)] \wedge \neg EG \neg\phi_2$$

(Exercise: Verify this!)

Fairness

A complicated subject.

Example:

- An arbiter is unfair if it ignores a request forever

One often wishes to assume a system is fair:

- The state transition relation may look as if it allows a path to be taken that ignores particular possibilities forever. . .
- But you **know** this is not actually possible.

Fairness Formally

Fix a set $F = \{f_1, f_2, \dots, f_n\}$, where each f_i is a set of states (a “fairness constraint”).

A path is **fair** if it includes states from each $f_i \in F$ infinitely often.

Alternatively

- π satisfies constraint f_i if $\{j \mid \pi_j \in f_i\}$ is infinite
- π is fair if it satisfies all constraints in F .

Decide we're only interested in paths that are **fair**.

This affects the definition of the **A** and **E** modalities.

Fair Semantics

$(W, R, V, F), w \models A[\phi_1 \cup \phi_2]$

if for every π such that $R\text{-path}(\pi, w)$ and π is **fair wrt** F , there exists an $i \geq 0$ such that $\mathfrak{M}, \pi(i) \models \phi_2$ and for all $0 \leq j < i$, $\mathfrak{M}, \pi(j) \models \phi_1$

$(W, R, V, F), w \models E[\phi_1 \cup \phi_2]$

if there exists a π such that $R\text{-path}(\pi, w)$ and π is **fair wrt** F and there also exists an $i \geq 0$ such that $\mathfrak{M}, \pi(i) \models \phi_2$ and for all $0 \leq j < i$, $\mathfrak{M}, \pi(j) \models \phi_1$

Similarly for AX and EX .

Can't We Do This in CTL?

Earlier example:

$AG(AF\textit{DeviceEnabled})$: The variable *DeviceEnabled* holds infinitely often on every computation path

Yes, we can require a model to have only fair paths.

But that's **not** the same thing as quantifying over only the fair paths, thereby ignoring non-fair paths.

CTL can't write:

$$A(\underbrace{G(AF\psi)}_{\text{path is fair}} \Rightarrow \phi)$$

(But you could write this in CTL*...)

Summary

- **CTL** is a rich logic for describing systems with branching structure
- **Syntax** combines quantification over paths (**A** and **E**), with quantification over states in those paths (**X**, **F**, **G**, and **U**)
- The **until** modalities (**A[$_ U _$]** and **E[$_ U _$]**) are important and powerful
- **Semantics** is in terms of
 - **worlds** (typical of a modal logic), and
 - **paths** (infinite sequences of worlds)
- **Fairness** is important in some applications
- (Two exercises!)