CTL vs. LTL

Robert Bellarmine Krug

Department of Computer Sciences
University of Texas at Austin

May 25, 2010

Outline

Outline

Kripke Structures — Definition

Let AP be a set of labels — i.e., a set of atomic propositions such as Boolean expressions over variables, constants, and predicate symbols.
A Kripke structure is a 4-tuple, $M = (S, I, R, L)$:

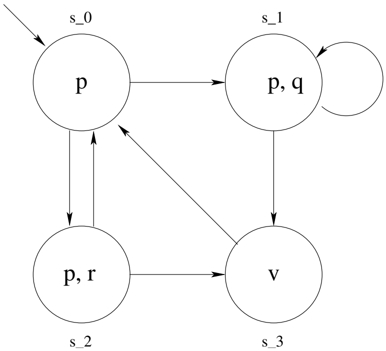
- a finite set of states, S,
- a set of initial states, $I \subseteq S$,
- a transition relation, $R \subseteq S \times S$ where $\forall s \in S, \exists s' \in S$ such that $(s, s') \in R$,
- a labeling function, L, from states to the power set of atomic propositions, $L : S \rightarrow 2^{AP}$.

Kripke Structure — An Example

$S = \{s_0, s_1, s_2, s_3\}$
$I = \{s_0\}$

$$
\begin{aligned}
R = \{ & \{s_0, \quad s_1\} \\
& \{s_0, \quad s_2\} \\
& \{s_1, \quad s_1\} \\
& \{s_1, \quad s_3\} \\
& \{s_2, \quad s_0\} \\
& \{s_2, \quad s_3\} \\
& \{s_3, \quad s_0\}\}
\end{aligned}
$$

$$
\begin{aligned}
L = \{ & \{s_0, \quad \{p\}\} \\
& \{s_1, \quad \{p, q\}\} \\
& \{s_2, \quad \{p, r\}\} \\
& \{s_3, \quad \{v\}\}\}
\end{aligned}
$$



Infinite Paths

LTL and CTL are concerned only with infinite paths. From here on, π will always denote an infinite path. Furthermore, π_0 will always denote π's first element, π_1 its second element, and so on.

$\pi = (\pi_0, \pi_1, \pi_2, \ldots)$ is an infinite path in M if it respects M's transition relation, i.e., $\forall i, (\pi_i, \pi_{i+1}) \in R$.

π^i denotes π's ith suffix, i.e., $\pi^i = (\pi_i, \pi_{i+1}, \pi_{i+2}, \ldots)$

$(\pi^i)^j = (\pi_i, \pi_{i+1}, \pi_{i+2}, \ldots)^j = (\pi_{i+j}, \pi_{i+j+1}, \pi_{i+j+2}, \ldots) = \pi^{i+j}$

Outline

LTL BNF Syntax

A well-formed LTL formula, ϕ, is recursively defined by the BNF formula:

$$
\begin{array}{rlll}
\phi ::= & \top & & ;\ \text{top, or true} \\
| & \bot & & ;\ \text{bottom, or false} \\
| & p & & ;\ p \text{ ranges over } AP \\
| & \neg\phi & & ;\ \text{negation} \\
| & \phi \wedge \phi & & ;\ \text{conjunction} \\
| & \phi \vee \phi & & ;\ \text{disjunction} \\
| & X\phi & & ;\ \text{next time} \\
| & F\phi & & ;\ \text{eventually} \\
| & G\phi & & ;\ \text{always} \\
| & \phi U \phi & & ;\ \text{until}
\end{array}
$$

From here on, lowercase letters such as p, q, and r, will denote atomic propositions. Greek letters such as ϕ and ψ will denote formulae.

LTL Semantics — the Basics

We now define the binary satisfaction relation, denoted by $\models$, for LTL formulae. This satisfaction is with respect a pair — $\langle M, \pi \rangle$, a Kripke structure and a path thereof.

First, the basics:

- $M, \pi \models \top$
  true is always satisfied
- $M, \pi \not\models \bot$
  false is never satisfied
- $(M, \pi \models p)$ if and only if $(p \in L(\pi_0))$
  atomic propositions are satisfied when they are members of the path's first element's labels

LTL Semantics — Boolean Combinations

The use of the Boolean operators $\neg$, $\wedge$, and $\vee$ in LTL formulae is a deliberate pun on their mathematical meanings.

- $(M, \pi \vDash \neg\phi)$ if and only if $(M, \pi \nvDash \phi)$
- $(M, \pi \vDash \phi \wedge \psi)$ if and only if $[(M, \pi \vDash \phi) \wedge (M, \pi \vDash \psi)]$
- $(M, \pi \vDash \phi \vee \psi)$ if and only if $[(M, \pi \vDash \phi) \vee (M, \pi \vDash \psi)]$

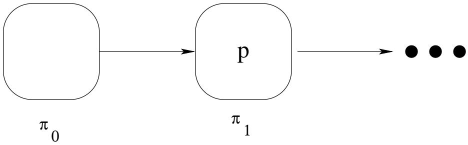
LTL Semantics — Temporal Operators

- $(M, \pi \models X\phi)$ if and only if $(M, \pi^1 \models \phi)$
  next time ϕ
- $(M, \pi \models F\phi)$ if and only if $(\exists i$ such that $M, \pi^i \models \phi)$
  eventually ϕ
- $(M, \pi \models G\phi)$ if and only if $(\forall i$ such that $M, \pi^i \models \phi)$
  always ϕ
- $(M, \pi \models \phi U \psi)$ if and only if
  $[\,\exists i$ such that $(\forall j < i(M, \pi^j \models \phi)) \wedge (M, \pi^i \models \psi)]$
  ϕ until ψ

N.B., The U used here is the "strong until." There is also a "weak until," $\phi U_w \psi$ is equivalent to $(\phi U \psi) \vee (G\phi)$.

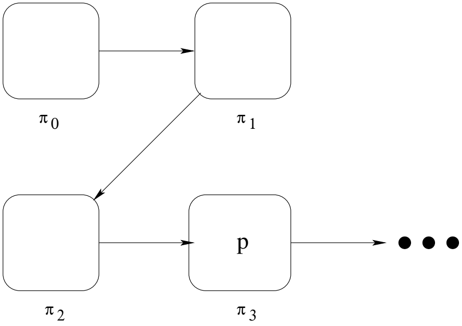
Xp — Example Path

$M, (\pi_0, \pi_1, \ldots) \models Xp$



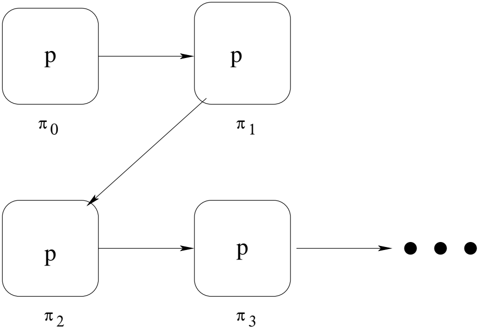
Fp — Example Path

$M, (\pi_0, \pi_1, \pi_2, \pi_3, \ldots) \vDash Fp$



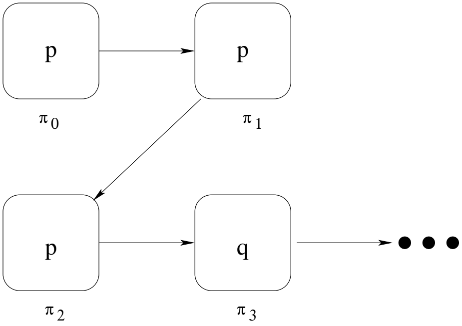
Gp — Example Path

$M, (\pi_0, \pi_1, \pi_2, \pi_3, \ldots) \models Gp$



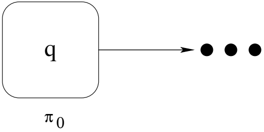
pUq — Example Path

$M, (\pi_0, \pi_1, \pi_2, \pi_3, \ldots) \vDash pUq$



pUq — Another Example Path

$M, (\pi_0, \ldots) \vDash pUq$



More LTL Semantics

- $(M \vDash_M \phi)$ if and only if $\forall \pi$ such that $\pi_0 \in I$, $(M, \pi \vDash \phi)$
  A model, or Kripke structure, satisfies an LTL formula, when all its paths do.
- $(\phi \equiv \psi)$ if and only if $\forall M \, [(M \vDash_M \phi) \Leftrightarrow (M \vDash_M \psi)]$
  Two LTL formulae are equivalent when they are satisfied by the same Kripke structures.

An LTL Equivalence

$X(\phi \land \psi) \equiv X\phi \land X\psi$

By the previous slide, this is true if, for all M and π:

$$[M, \pi \vDash X(\phi \land \psi)] \Leftrightarrow [M, \pi \vDash (X\phi \land X\psi)]$$

$[M, \pi \vDash X(\phi \land \psi)] =$

$[M, \pi^1 \vDash (\phi \land \psi)] =$	by definition of X
$[(M, \pi^1 \vDash \phi) \land (M, \pi^1 \vDash \psi)] =$	by definition of $\land$
$[(M, \pi \vDash X\phi) \land (M, \pi \vDash X\psi)] =$	by definition of X
$[M, \pi \vDash (X\phi \land X\psi)]$	by definition of $\land$

Some More LTL Equivalences

$X(\phi \wedge \psi) \equiv X\phi \wedge X\psi$
$X(\phi \vee \psi) \equiv X\phi \vee X\psi$
$X(\phi U \psi) \equiv (X\phi U X\psi)$
$\neg X\phi \equiv X\neg\phi$

$F(\phi \vee \psi) \equiv F\phi \vee F\psi$
$G(\phi \wedge \psi) \equiv G\phi \wedge G\psi$
$\neg F\phi \equiv G\neg\phi$

$(\phi \wedge \psi) U \rho \equiv (\phi U \rho) \wedge (\psi U \rho)$
$\rho U(\phi \vee \psi) \equiv (\rho U \phi) \vee (\rho U \psi)$

$FF\phi \equiv F\phi$
$GG\phi \equiv G\phi$

Outline

1. Some Definitions And Notation

2. LTL

3. CTL

4. CTL vs. LTL

CTL BNF Syntax

A well-formed CTL formula, ϕ, is recursively defined by the BNF formula (N.B., AX, AF, etc., are each single symbols, not pairs of symbols):

$$
\begin{array}{rll}
\phi ::= & \top & \\
| & \bot & \\
| & p & \\
| & \neg\phi & \\
| & \phi \wedge \phi & \\
| & \phi \vee \phi & \\
| & AX\phi & ;\ A \text{ — for all paths} \\
| & AF\phi & \\
| & AG\phi & \\
| & \phi AU \phi & \\
| & EX\phi & ;\ E \text{ — there exists a path} \\
| & EF\phi & \\
| & EG\phi & \\
| & \phi EU \psi &
\end{array}
$$

CTL Semantics — the Basics

As for LTL, we now define the satisfaction relation. Again, this satisfaction is with respect to a pair, but this time $\langle M, s \rangle$, a Kripke structure and a state thereof. This change from path to state creates a very different logic.

- $M, s \models \top$
- $M, s \not\models \bot$
- $(M, s \models p)$ if and only if $(p \in L(s))$
  atomic propositions are satisfied when they are members of the state's labels

CTL Semantics — Boolean Combinations

As for LTL, the use of the Boolean operators $\neg$, $\wedge$, and $\vee$ in CTL formulae is a deliberate pun on their mathematical meanings.

- $(M, s \models \neg\phi)$ if and only if $(M, s \not\models \phi)$
- $(M, s \models \phi \wedge \psi)$ if and only if $((M, s \models \phi) \wedge (M, s \models \psi))$
- $(M, s \models \phi \vee \psi)$ if and only if $((M, s \models \phi) \vee (M, s \models \psi))$

CTL Semantics — Temporal Operators, the A team

- $(M, s \models AX\phi)$ if and only if $(\forall \pi \text{ such that } \pi_0 = s, M, \pi^1 \models \phi)$
  for all paths starting at s, next time ϕ
- $(M, s \models AF\phi)$ if and only if
  $(\forall \pi \text{ such that } \pi_0 = s, \exists i \text{ such that } M, \pi^i \models \phi)$
  for all paths starting at s, eventually ϕ
- $(M, s \models AG\phi)$ if and only if
  $(\forall \pi \text{ such that } \pi_0 = s, \forall i\ M, \pi^i \models \phi)$
  for all paths starting at s, always ϕ
- $(M, s \models \phi AU\psi)$ if and only if
  $(\forall \pi \text{ such that } \pi_0 = s, \exists i \text{ such that}$
  $(\forall j < i(M, \pi^j \models \phi)) \wedge (M, \pi^i \models \psi))$
  for all paths starting at s, ϕ until ψ

CTL Semantics — Temporal Operators, the E team

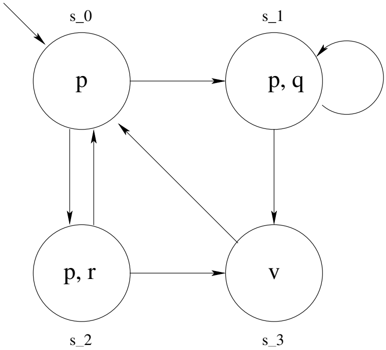
- $(M, s \vDash EX\phi)$ if and only if $(\exists\pi$ such that $\pi_0 = s, M, \pi^1 \vDash \phi)$
  there exists a path such that next time ϕ
- $(M, s \vDash EF\phi)$ if and only if
  $(\exists\pi$ such that $\pi_0 = s, \exists i$ such that $M, \pi^i \vDash \phi)$
  there exists a path such that eventually ϕ
- $(M, s \vDash EG\phi)$ if and only if
  $(\exists\pi$ such that $\pi_0 = s, \forall i\ M, \pi^i \vDash \phi)$
  there exists a path such that always ϕ
- $(M, s \vDash \phi EU\psi)$ if and only if
  $(\exists\pi$ such that $\pi_0 = s, \exists i$ such that
  $(\forall j < i(M, \pi^j \vDash \phi)) \wedge (M, \pi^i \vDash \psi))$
  there exists a path such that ϕ until ψ

AXp

$S = \{s_0, s_1, s_2, s_3\}$
$I = \{s_0\}$

$M, s_0 \models AXp$

$$R = \{\{s_0, \quad s_1\} \\ \{s_0, \quad s_2\} \\ \{s_1, \quad s_1\} \\ \{s_1, \quad s_3\} \\ \{s_2, \quad s_0\} \\ \{s_2, \quad s_3\} \\ \{s_3, \quad s_0\}\}$$



$$L = \{\{s_0, \quad \{p\}\} \\ \{s_1, \quad \{p, q\}\} \\ \{s_2, \quad \{p, r\}\} \\ \{s_3, \quad \{v\}\}\}$$

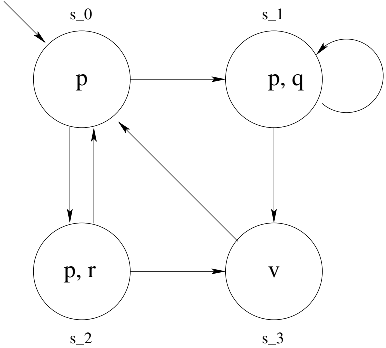
EFv

$S = \{s_0, s_1, s_2, s_3\}$
$I = \{s_0\}$

$$
\begin{aligned}
R = \{&\{s_0, & s_1\} \\
&\{s_0, & s_2\} \\
&\{s_1, & s_1\} \\
&\{s_1, & s_3\} \\
&\{s_2, & s_0\} \\
&\{s_2, & s_3\} \\
&\{s_3, & s_0\}\}
\end{aligned}
$$

$$
\begin{aligned}
L = \{&\{s_0, & \{p\}\} \\
&\{s_1, & \{p, q\}\} \\
&\{s_2, & \{p, r\}\} \\
&\{s_3, & \{v\}\}\}
\end{aligned}
$$

$M, s_0 \models EFv$



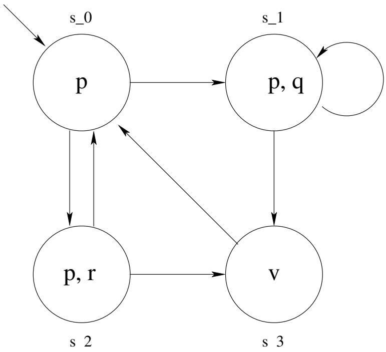
$AG(p \vee v)$

$S = \{s_0, s_1, s_2, s_3\}$
$I = S$

$M, s_0 \models AG(p \vee v)$

$$
\begin{aligned}
R = \{&\{s_0, \quad s_1\}\\
&\{s_0, \quad s_2\}\\
&\{s_1, \quad s_1\}\\
&\{s_1, \quad s_3\}\\
&\{s_2, \quad s_0\}\\
&\{s_2, \quad s_3\}\\
&\{s_3, \quad s_0\}\}
\end{aligned}
$$

$$
\begin{aligned}
L = \{&\{s_0, \quad \{p\}\}\\
&\{s_1, \quad \{p, q\}\}\\
&\{s_2, \quad \{p, r\}\}\\
&\{s_3, \quad \{v\}\}\}
\end{aligned}
$$



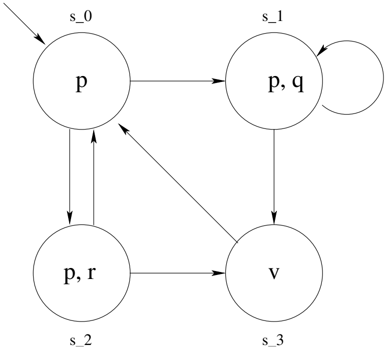
$pEUv$

$$S = \{s_0, s_1, s_2, s_3\}$$
$$I = S$$

$$M, s_0 \vDash pEUv$$

$$R = \{\{s_0, \quad s_1\}$$
$$\{s_0, \quad s_2\}$$
$$\{s_1, \quad s_1\}$$
$$\{s_1, \quad s_3\}$$
$$\{s_2, \quad s_0\}$$
$$\{s_2, \quad s_3\}$$
$$\{s_3, \quad s_0\}\}$$

$$L = \{\{s_0, \quad \{p\}\}$$
$$\{s_1, \quad \{p, q\}\}$$
$$\{s_2, \quad \{p, r\}\}$$
$$\{s_3, \quad \{v\}\}\}$$



More CTL Semantics

- $(M \vDash_M \phi)$ if and only if $\forall s \in I, (M, s \vDash \phi)$
  A model, or Kripke structure, satisfies a CTL formula, when all its states do.
- $(\phi \equiv \psi)$ if and only if $\forall M \; [(M \vDash_M \phi) \Leftrightarrow (M \vDash_M \psi)]$
  Two CTL formulae are equivalent when they are satisfied by the same Kripke structures.

Some CTL Equivalences

$AX(\phi \wedge \psi) \equiv AX\phi \wedge AX\psi$
$EX(\phi \vee \psi) \equiv EX\phi \vee EX\psi$
$\neg AX\phi \equiv EX\neg\phi$

$EF(\phi \vee \psi) \equiv EF\phi \vee EF\psi$
$AG(\phi \wedge \psi) \equiv AG\phi \wedge AG\psi$
$\neg AF\phi \equiv EG\neg\phi$
$\neg EF\phi \equiv AG\neg\phi$

$AFAF\phi \equiv AF\phi$
$EFEF\phi \equiv EF\phi$
$AGAG\phi \equiv AG\phi$
$EGEG\phi \equiv EG\phi$

Outline

Complexity

$|\phi| = n$, $|M| = m$

CTL: $O(mn)$

LTL: $O(m2^n)$ — (and PSpace complete)

Intuitiveness

IBM Journal or Research and Development: Formal Verification Made Easy, 1997

> *We found only simple CTL equations to be comprehensible; nontrivial equations are hard to understand and prone to error.*

CAV'98: On the Fly Model Checking, 1998

> *CTL is difficult to use for most users and requires a new way of thinking about hardware.*

LTL and CTL Equivalence

A CTL formula ϕ_{CTL} and an LTL formula ϕ_{LTL} are equivalent if they are satisfied by the same Kripke structures:

$\phi_{CTL} \equiv \phi_{LTL}$ if and only if $[(M \vDash_M \phi_{CTL}) \Leftrightarrow (M \vDash_M \phi_{LTL})]$

E

Any CTL formula necessitating E cannot be expressed in LTL.

Example: EXp

G

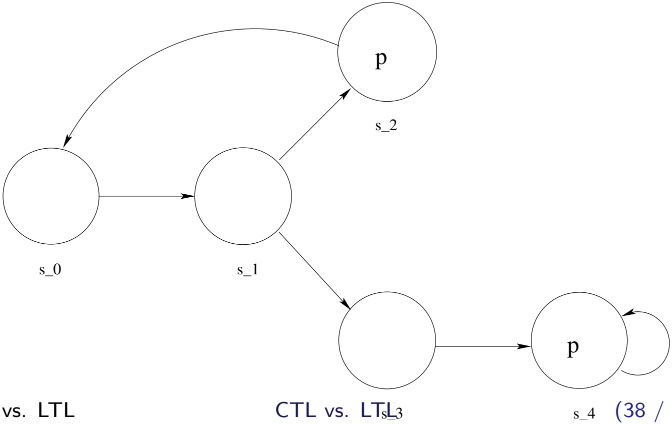
For any CTL formula ϕ_{CTL} and LTL formula ϕ_{LTL} such that $\phi_{CTL} \equiv \phi_{LTL}$,

$AG\phi_{CTL} \equiv G\phi_{LTL}$

$AFAXp$

$FXp \equiv XFp \equiv AXAFp \not\equiv AFAXp$

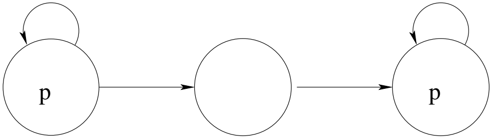
The below example satisfies $AXAFp$, but not $AFAXp$. The latter of these says that, starting in any state, along all paths we will eventually reach a state, all of whose immediate successors satisfy p.



$AFAGp$

$FGp \not\equiv AFAGp$

The below example satisfies FGp, but not $AFAGp$. The latter says that starting in any state, along all paths we will eventually reach a part of the model from which all successors satisfy p. But consider the path cycling through s_0 — then s_1 will always be a potential successor.



$GFp \Rightarrow GFq$

$(GFp \equiv AGAFp)$, but $(GFp \Rightarrow GFq) \not\equiv (AGAFp \Rightarrow AGAFq)$

While $GFp \equiv AGAFp$, the above implications are not equivalent.

The LTL formula is an implication about paths, but the two parts of the CTL formula determine subsets of states independantly. The below example satisfies $AGAFp \Rightarrow AGAFq$ but not $GFp \Rightarrow GFq$. The CTL is trivially satisfied, because $AGAFp$ is not satisfied. The LTL is not satisfied, because the path cycling through s_0 forever satisfies GFp but not GFq.