CIS 6930/4930 Computer Aided Verification

Temporal Logics

Hao Zheng
Dept. of Computer Science & Eng.
Univ. of South Florida
Temporal Logics

- A formalism to describes properties of paths in the model of reactive systems.
- First order logic augmented with temporal operators.
- Time is implicit.
  - Explicit in real-time temporal logics.
- There exist different temporal logics.
  - With different view of underlying computation.
- CTL* (CTL) views computation of a system as a tree.
  - System can move into different future.
- LTL views computation of a system as a set of paths.
  - System has only one direction into the future.
Models of Computations

State transition graph
Kripke Structure

Computation tree

Paths

...
Computational Tree Logic CTL*

- To describe paths from a given state.
- Path quantifiers:
  - $A$: for all computation paths from a state.
  - $E$: for some computation path(s) from a state.
- Linear temporal operators: describe properties along a path.
  - $Gp \rightarrow p$ holds in every state on the path.
  - $Fp \rightarrow p$ holds in some state on the path.
  - $Xp \rightarrow p$ holds in the second state of the path
  - $pUq \rightarrow p$ holds until $q$ holds in some state on the path.
  - $pWq$ – similar to $U$, but $q$ does not need to hold.
State and Path Formulas

- Path formulas hold along a path.
  - If $f$ is a state formula, it is also a path formula.
  - If $f$ and $g$ are path formulas, so are boolean combinations of $f$ and $g$, $\text{X} f$, $\text{F} f$, $\text{G} f$, and $f \text{U} g$.

- State formulas hold at a state.
  - If $p$ is an atomic proposition, then $p$ is a state formula.
  - If $f$ and $g$ are state formulas, so are boolean combinations of $f$ and $g$.
  - If $f$ is a path formula, $\text{A} f$ and $\text{E} f$ are state formulas.

- CTL* formulae are state formulas generated by the above rules.
Semantics: Path Formulas

• Defined w.r.t a Kripke structure $M$.
• If $f$ is a path formula, $M, \pi \models f$ means $f$ holds along path $\pi$.
• Definitions:
  - $M, \pi \models f \iff f$ is a state formula, $s$ is the first state of $M$, $s \models f$ holds if $p$ is an atomic proposition and $p \in L(s)$.
  - $M, \pi \models \neg f \iff f$ is a path formula, and $M, \pi \models f$ does not hold.
  - $M, \pi \models f \lor g \iff f$ and $g$ are path formulas, and $M, s \models f$ or $M, s \models f$.
  - $M, \pi \models f \land g \iff f$ and $g$ are path formulas, and $M, s \models f$ and $M, s \models f$.
  - $M, \pi \models X f \iff f$ is a path formula, and $M, \pi^1 \models f$.
  - $M, \pi \models F f \iff f$ is a path formula, and $M, \pi^k \models f$ for some $k \geq 0$.
  - $M, \pi \models G f \iff f$ is a path formula, and $M, \pi^k \models f$ for all $k \geq 0$.
  - $M, \pi \models f U g \iff ...$
Semantics: State Formulas

- Defined w.r.t a Kripke structure $M$.
- If $f$ is a state formula, $M, s \models f$ means $f$ holds at state $s$ of $M$.
- Definitions:
  - $M, s \models p \iff$ if $p$ is an atomic proposition and $p \in L(s)$.
  - $M, s \models \neg f \iff M, s \not\models f$ does not hold.
  - $M, s \models f \lor g \iff M, s \models f$ or $M, s \models f$.
  - $M, s \models f \land g \iff M, s \models f$ and $M, s \models f$.
  - $M, s \models A f \iff f$ is a path formulas, and for all paths $\pi$ from $s$ such that $M, \pi \models f$.
  - $M, s \models E f \iff f$ is a path formula, and there is a path $\pi$ from $s$ such that $M, \pi \models f$. 
Equivalences

• Not all operators are essential to express a property.
  - \( f \land g \equiv \neg(\neg f \lor \neg g) \)
  - \( A f \equiv \neg E (\neg f) \)
  - \( G f \equiv \neg F (\neg f) \)
  - \( F f \equiv (true \cup f) \)
  - \( F(f \lor g) \equiv Ff \lor Fg \)
    - What about \( F(f \land g) \equiv Ff \land Fg \)?
  - \( G(f \land g) \equiv Gf \land Gg \)
    - What about \( G(f \lor g) \equiv Gf \lor Gg \)?
CTL and LTL

• CTL* is more expressive, but expensive for verification.
• Two useful sublogics of CTL*: CTL and LTL.
• CTL is a restricted subset of CTL* where temporal operators must be immediately preceded by a path quantifier.
  – Basic operators: $AG, AF, AX, A(U), EG, EF, EX, E(U)$.
  – Example: $AG( EF f)$
• LTL consists of formulas of the form $Af$ defined as follows:
  – If $p$ is an atomic formula, the $p$ is a path formula.
  – if $f$ and $g$ are path formulas, so are boolean combinations of $f$ and $g$, $Xf, Ff, Gf$, and $f U g$.
  – Example: $A( FG f)$
Interpretation of CTL Operators

\[ AG_f \text{ is true} \]

\[ EG_f \text{ is true} \]
Interpretation of CTL Operators

$AF \phi$ is true

$EF \phi$ is true
Interpretation of CTL Operators

$\text{AX} f$ is true

$\text{EX} f$ is true
Interpretation of CTL Operators

A(\(f\ U\ g\)) is true

E(\(f\ U\ g\)) is true
A Sufficient Set of CTL Operators

• Any CTL formulas can be expressed using \( \text{EX} \), \( \text{EG} \), and \( \text{EU} \).
  
  \[ \begin{align*}
  \text{AX} f &= \neg \text{EX} \neg f \\
  \text{AG} f &= \neg \text{EF} \neg f = \neg \text{E}( \text{true} \ U \neg f) \\
  \text{AF} f &= \neg \text{EG} \neg f \\
  \text{A} (f \ U g) &= (\neg \text{EG} \neg g) \land (\neg \text{E} (\neg g \ U (\neg f \land \neg g)) \\
  \text{What does } \text{AG}(\text{AF} f) \text{ mean?}
  \end{align*} \]
LTL Semantics Example

\[ M, s_0 \models p \land q \]
\[ M, s_0 \models \mathbf{X} r \]
\[ M, s_0 \models \mathbf{G} \neg(p \land r) \]
\[ M, s_0 \models \mathbf{G} (\mathbf{F} p) \]
A Sufficient Set of LTL Operators

• \{U, X\}, \{R, X\}, or \{W, X\} is sufficient.
  – \(Gf \equiv \neg F \neg f\)
  – \(\neg Xf \equiv X \neg f\)
  – \(f R g \equiv \neg (\neg f \ U \ \neg g)\)
  – \(f U g \equiv f \ W g \land Fg\)
  – \(Ff \equiv \text{true } U f\)

• Examples: \(GFf\) and \(FGf\)?
CTL*, CTL, and LTL

- CTL formula specifies a set of states.
- A LTL formula specifies a set of paths.
- $\text{AG}(\text{EF}\ f)$ is not expressible in CTL.
- $\text{AG(EF}\ f)$ is not expressible in LTL.
Safety and Liveness Properties

- Safety: nothing bad should happen.
- Liveness: something good eventually happens.
- Example: a mutual exclusion element.
Fairness

• Fairness means certain properties happen infinitely often during computation.
  – An arbiter cannot ignore some requests forever.
  – A communication channel cannot lose message all the time.

• Models may contain unfair computations.
  – Non-deterministic models of physical computating systems.
  – Wrong implementations of fairness requirements.

• Fairness constraints eliminate the unfair computations.
  – Unfairness introduced to simplify modeling.

• Fair computations satisfy fairness constraints infinitely often.
Fair Semantics

- Fairness constraints are expressed as sets of states that hold infinitely often on computations.

- CTL* semantics with fairness
  - \( M, s \models_{F} p \leftrightarrow \) if there is a fair path from \( s \) and \( p \in L(s) \).
  - \( M, s \models_{F} A f \leftrightarrow f \) is a path formulas, and for all fair paths \( \pi \) from \( s \) such that \( M, \pi \models f \).
  - \( M, s \models_{F} E f \leftrightarrow f \) is a path formula, and there is a fair path \( \pi \) from \( s \) such that \( M, \pi \models f \).

- CTL – will discuss it later.

- LTL – fairness can be easily expressed and incorporated with verification.
  - Ex.: \( GF p \), or \( GF p \rightarrow GF q \)