SE 2F03 Fall 2005

04 Temporal Logic and Model Checking

Instructor: W. M. Farmer

Revised: 8 November 2005

What is Verification?

- **Verification** is the process of checking whether an **implementation** of a system satisfies the **requirements** of the system.

- Verification is usually impractical without computer support for:
  - Writing the documentation that the describes the requirements and implementation of the system.
  - Performing the verification.

- Verification can be applied at different levels, from high-level design to actual code.
  - Usually verification is more effective at a high level of abstraction, while testing is more effective at a low level.

Proof-Based Verification

- A system is specified by a theory T in a logic and a requirement for the system is specified by a formula R.

- $T \models R$ means that an implementation of the system satisfies the requirement specified by R.

- $T \models R$ is verified by showing $T \vdash_{\mathbf{P}} R$ for some sound proof system **P** for T.
  - The verification method consists of the user trying to **interactively prove** R from T in **P**.

Specifying a System as a Theory

Three ways of that a system can be specified as a theory:

1. T specifies that the system is a relation between **inputs and outputs**, and a model of T is a function from inputs to outputs.

2. T specifies that the system is a relation between **before states and after states**, and a model of T is a function from states to states.

3. T specifies that the system is a relation between **before histories and after histories**, and a model of T is a function from histories to histories.

Model-Based Verification

- A system is specified by a model M for a logic and a requirement for the system is specified by a formula R.
  - For example, M can be a finite state machine and R a formula in a temporal logic.

- $M \models R$ means that an implementation of the system satisfies the requirement specified by R.

- $M \models R$ is verified by showing that R is true in M.
  - The verification method consists of trying to **automatically compute** whether $M \models R$ holds.

Temporal Logic

- A **temporal logic** is a logic in which the value of an expression can depend on **time**.
  - There are many different flavors of temporal logic.

- A key attribute of a temporal logic is how time is represented.
  - Time can be represented as **linear** or **branching**.
  - Time can be represented as **continuous** or **discrete**.

Temporal Logics for Model Checking

- **Linear-time Temporal Logic (LTL)** is a temporal logic where time is linear and discrete.
  - Implicitly quantifies over all paths through time.

- **Computation Temporal Logic (CTL)** is a temporal logic where time is branching and discrete.
  - Allows explicit quantification over paths through time.

Syntax of LTL

- A **language** of LTL is a set L of **atoms** (**propositional symbols)**.
  - Each atom represents an atomic proposition.

- A **formula** of L is a string of symbols inductively defined by the following formation rules:
  - Each $p \in L$ is a formula of L.
  - $\top$ and $\bot$ are formulas of L.
  - If φ and ψ are formulas of L, then $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, and $(\varphi \rightarrow \psi)$ are formulas of L.
  - If φ and ψ are formulas of L, then, $(\mathsf{X}\ \varphi)$, $(\mathsf{F}\ \varphi)$, $(\mathsf{G}\ \varphi)$, $(\varphi\ \mathsf{U}\ \psi)$, $(\varphi\ \mathsf{W}\ \psi)$, and $(\varphi\ \mathsf{R}\ \psi)$ are formulas of L.

- X, F, G, U, R, and W are **temporal connectives**.

Transition Systems

- A **transition system** is a pair $M = (S, \rightarrow)$ such that:
  - S is a set of **states**.
  - $\rightarrow$ is a binary relation on S such that, for all $s \in S$, there is some $s' \in S$ with $s \rightarrow s'$.

- When S is finite, $(S, \rightarrow)$ is a special case of a **finite state machine**.
  - Finite state machines may also have inputs, outputs, and designated start and final states.

- A **path** in a transition system $(S, \rightarrow)$ is an infinite sequence $\pi = s_1, s_2, s_3, \ldots$ of states in S such that $s_i \rightarrow s_{i+1}$ for all $i \geq 1$.
  - π may be written as $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \cdots$.
  - For $i \geq 1$, π^i is the path $s_i \rightarrow s_{i+1} \rightarrow s_{i+2} \rightarrow \cdots$.

Semantics of LTL: Models

- Let L be a language for LTL.

- A **model** for L is a triple $M = (S, \rightarrow, I)$ where:
  - $(S, \rightarrow)$ is a transition system.
  - I is an (interpretation) function that assigns a truth value in $\{\mathsf{t}, \mathsf{f}\}$ to each $(s, p) \in S \times L$.

Semantics of LTL: Valuation Function (1)

The **valuation function** for a model $M = (S, \rightarrow, I)$ for L is the binary function V that satisfies the following conditions for all paths $\pi = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \cdots$ in M and all formulas φ of L:

1. Let $\varphi \in L$. Then $V(\pi, \varphi) = I(s_1, \varphi)$.

2. Let $\varphi = \top$. Then $V(\pi, \varphi) = \mathsf{t}$.

3. Let $\varphi = \bot$. Then $V(\pi, \varphi) = \mathsf{f}$.

4. Let $\varphi = \neg\varphi'$. Then $V(\pi, \varphi) = \mathsf{t}$ iff $V(\pi, \varphi') = \mathsf{f}$.

5. Let $\varphi = \varphi_1 \wedge \varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff $V(\pi, \varphi_1) = \mathsf{t}$ and $V(\pi, \varphi_2) = \mathsf{t}$.

6. Let $\varphi = \varphi_1 \vee \varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff $V(\pi, \varphi_1) = \mathsf{t}$ or $V(\pi, \varphi_2) = \mathsf{t}$.

7. Let $\varphi = \varphi_1 \rightarrow \varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff $V(\pi, \varphi_1) = \mathsf{t}$ implies $V(\pi, \varphi_2) = \mathsf{t}$.

Semantics of LTL: Valuation Function (2)

8. Let $\varphi = \mathsf{X}\,\varphi'$. Then $V(\pi, \varphi) = \mathsf{t}$ iff $V(\pi^2, \varphi') = \mathsf{t}$.

9. Let $\varphi = \mathsf{F}\,\varphi'$. Then $V(\pi, \varphi) = \mathsf{t}$ iff, for some $i \geq 1$, $V(\pi^i, \varphi') = \mathsf{t}$.

10. Let $\varphi = \mathsf{G}\,\varphi'$. Then $V(\pi, \varphi) = \mathsf{t}$ iff, for all $i \geq 1$, $V(\pi^i, \varphi') = \mathsf{t}$.

11. Let $\varphi = \varphi_1\,\mathsf{U}\,\varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff, for some $i \geq 1$, $V(\pi^i, \varphi_2) = \mathsf{t}$ and, for all $j = 1, \ldots, i-1$, $V(\pi^j, \varphi_1) = \mathsf{t}$.

12. Let $\varphi = \varphi_1\,\mathsf{W}\,\varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff either, for some $i \geq 1$, $V(\pi^i, \varphi_2) = \mathsf{t}$ and, for all $j = 1, \ldots, i-1$, $V(\pi^j, \varphi_1) = \mathsf{t}$ or, for all $i \geq 1$, $V(\pi^i, \varphi_1) = \mathsf{t}$.

13. Let $\varphi = \varphi_1\,\mathsf{R}\,\varphi_2$. Then $V(\pi, \varphi) = \mathsf{t}$ iff either, for some $i \geq 1$, $V(\pi^i, \varphi_1) = \mathsf{t}$ and, for all $j = 1, \ldots, i$, $V(\pi^j, \varphi_2) = \mathsf{t}$ or, for all $i \geq 1$, $V(\pi^i, \varphi_2) = \mathsf{t}$.

Semantics of LTL: Satisfiability

- Let L be a language of LTL, M a model for L, V the valuation function for M, π a path in M, s a state of M, and φ a formula of L.

- $\pi \models \varphi$ means $V(\pi, \varphi) = \mathsf{t}$.

- $M, s \models \varphi$ means $\pi \models \varphi$ for every path π in M that starts at s.

- $M \models \varphi$ means $\pi \models \varphi$ for every path π in M.

Equivalences Between LTL Formulas

- Equivalences between duals:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi, \quad \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$

$$\neg \mathsf{G}\ \varphi \equiv \mathsf{F}\ \neg\varphi, \quad \neg \mathsf{F}\ \varphi \equiv \mathsf{G}\ \neg\varphi, \quad \neg \mathsf{X}\ \varphi \equiv \mathsf{X}\ \neg\varphi$$

$$\neg(\varphi\ \mathsf{U}\ \psi) \equiv \neg\varphi\ \mathsf{R}\ \neg\psi, \quad \neg(\varphi\ \mathsf{R}\ \psi) \equiv \neg\varphi\ \mathsf{U}\ \neg\psi$$

- Distribution laws:

$$\mathsf{F}\ (\varphi \vee \psi) \equiv \mathsf{F}\ \varphi \vee \mathsf{F}\ \psi, \quad \mathsf{G}\ (\varphi \wedge \psi) \equiv \mathsf{G}\ \varphi \wedge \mathsf{G}\ \psi$$

- Definition of F and G:

$$\mathsf{F}\ \varphi \equiv \top\ \mathsf{U}\ \varphi, \quad \mathsf{G}\ \varphi \equiv \bot\ \mathsf{R}\ \varphi$$

- U vs. W:

$$\varphi\ \mathsf{U}\ \psi \equiv (\varphi\ \mathsf{W}\ \psi) \wedge \mathsf{F}\ \psi, \quad \varphi\ \mathsf{W}\ \psi \equiv (\varphi\ \mathsf{U}\ \psi) \vee \mathsf{G}\ \varphi$$

Expressibility of LTL

- A set S of temporal connectives is **complete** or **adequate** for LTL if, for every formula of LTL, there is an equivalent formula that only uses the temporal connectives in S.
  - For example, each of $\{\mathsf{X}, \mathsf{U}\}$, $\{\mathsf{X}, \mathsf{R}\}$, $\{\mathsf{X}, \mathsf{W}\}$ is complete for LTL.
- Many temporal statements cannot be expressed in LTL.
- A statement that asserts the existence of certain path cannot be expressed in LTL, but can be expressed in CTL.
  - For example, "it is possible to use the dryer to dry clothes".
- A statement that quantifies over time cannot be (easily) expressed in LTL.
  - For example, "it will take twice the time to accomplish Task 2 than it takes to accomplish Task 1".

Model Checking

- There are tools that allow the user to:
  1. Specify a software system as a model M for a temporal logic.
  2. Specify a requirement for M starting at a state s as a formula φ in the temporal logic.
  3. Run a model checker that determines whether $M, s \models \varphi$.

- Model checking strategy for LTL:
  1. Construct an automaton $A_{\neg\varphi}$ such that, for all paths π, $\pi \models \neg\varphi$ iff the trace of π is accepted by $A_{\neg\varphi}$.
  2. Combine $A_{\neg\varphi}$ with the model M.
  3. Determine whether there is a path in M starting at state s whose trace is accepted by $A_{\neg\varphi}$. If there is no such path, $M, s \models \varphi$ is true. Otherwise, if there is such a path, $M, s \models \varphi$ is false and the path is a counterexample.