Reactive Systems: Temporal Logic $LTL$, part II

Timo Latvala

February 11, 2004
Semantics in a Kripke Structure

The semantics in a Kripke structure can be (equivalently) defined as:

**Definition 1** An $LTL$ formula $f$ holds in a Kripke structure $M$, denoted $M \models f$, iff for all paths $\pi$ in $M$, such that $\pi_0 = s^0$, it holds that $\pi \models f$.

We also get the following:

**Definition 2** An $LTL$ formula does not hold $f$ in a Kripke structure $M$, denoted $M \not\models f$, iff there is a path $\pi$ in $M$, such that $\pi_0 = s^0$ and $\pi \models \neg f$.

This definition is easier to handle, since to prove that $f$ does not hold, we only have to find one path where $\neg f$ holds. If such a path cannot be found, then we can conclude that $f$ holds.

©2003 Keijo Heljanko, ©2004 Timo Latvala
Universal Quantification

Note: It is possible that both $M \not\models f$ and $M \not\models \neg f$ hold.

Consider the formula $\Box a$ in the Kripke structure $M$ below. We get that $M \not\models \Box a$. However, $\neg \Box a = \Diamond \neg a$, and thus it is also the case that $M \not\models \neg \Box a$. 

©2003 Keijo Heljanko, ©2004 Timo Latvala
Example: Kripke Structure $M$ of a Mutex System
Example: Properties of the Mutex System

- $M \models \square \neg (CR0 \land CR1)$
  (both processes are never in their critical sections at the same time)

- $M \models \square (TRY0 \Rightarrow \Diamond CR0)$
  (always when process 0 enters the trying section, it is eventually followed by process 0 entering the critical section)

- $M \models \square (TRY0 \Rightarrow (TRY0 U CR0))$
  (always when process 0 enters the trying section, it stays in the trying section until it enters the critical section)
Example: Unsatisfied Properties of the Mutex System

- $M \not\models \diamond CR0$
  (the following does not hold: process 0 will eventually enter the critical section)

- $M \not\models \Box \diamond CR0 \Rightarrow \Box \diamond CR1$
  (the following does not hold: if process 0 is infinitely often in the critical section, then also process 1 is infinitely often in the critical section)
Example: More Properties of the Mutex System

- $M \models □♦CR0 \Rightarrow □♦TRY0$
  (if process 0 is infinitely often in the critical section, then it is also infinitely often in the trying section)

- $M \models ◊□NC0 \Rightarrow ◊□¬CR0$
  (if process 0 all the time from a certain point onward is in the non-critical section, then process 0 will all the time from a certain point onward be not in the critical section)

- $M \models (◊TRY0 \land ◊TRY1) \Rightarrow (◊CR0 \land ◊CR1)$
  (if both process 0 and 1 are infinitely often in the trying section, then both process 0 and 1 are infinitely often also in the critical section)
LTL Property Patterns: Scope

Quite often the requirements of a system follow some simple patterns. Sometimes we want to specify that a property should only hold in a certain context, called the scope of a property.

- **Global**: The scope of the property is the path.

- **Before** $R$: The scope of the property is all indexes which are strictly smaller than the first appearance of $R$.

- **After** $Q$: The scope of the property are all indexes greater or equal to the first appearance of $Q$. 

©2003 Keijo Heljanko, ©2004 Timo Latvala
\textbf{LTL Property Patterns: Scope}

- \textit{Between $Q$ and $R$}: The scope of the requirement contains all sequences of indexes, such that $Q$ holds at the first index until (but not including) the larger index where $R$ holds at the first time.

- \textit{After $Q$ until $R$}: The scope of the requirement contains all sequences of indexes, such that:
  \begin{itemize}
    \item [(i)] $Q$ holds at the first index until (but not including) the larger index where $R$ holds at the first time, or
    \item [(ii)] $Q$ holds at the first index but the $R$ never holds at a larger index.
  \end{itemize}

Note: Scopes are defined in a way that always includes the index at which the event triggering the scope happens, but excluding the index at which the event ending the scope happens.

©2003 Keijo Heljanko, ©2004 Timo Latvala
**LTL** Property Patterns: Scope

- **Global**
- **Before** $R$
- **After** $Q$
- **Between** $Q$ and $R$
- **After** $Q$ until $R$

©2003 Keijo Heljanko, ©2004 Timo Latvala
Absence pattern specifies that “P is false” should hold within the scope:

- **Global:** $\Box(\neg P)$
- **Before $R$:** $(\Diamond R) \Rightarrow (\neg P U R)$
- **After $Q$:** $\Box(Q \Rightarrow (\Box(\neg P)))$
- **Between $Q$ and $R$:** $\Box((Q \land \neg R \land \Diamond R) \Rightarrow (\neg P U R))$
- **After $Q$ until $R$:** $\Box((Q \land \neg R) \Rightarrow (\neg P U (R \lor \Box(\neg P))))$

©2003 Keijo Heljanko, ©2004 Timo Latvala
$LTL$ Property Patterns: Existence

Existence pattern specifies that “$P$ becomes true” within the scope:

- Global: $\Diamond P$

- Before $R$: $\neg R U ((P \land \neg R) \lor (\Box \neg R))$

- After $Q$: $(\Box (\neg Q)) \lor (\Diamond (Q \land (\Diamond P)))$

- Between $Q$ and $R$: $\Box ((Q \land \neg R) \Rightarrow (\neg R U ((P \land \neg R) \lor (\Box (\neg R))))))$

- After $Q$ until $R$: $\Box ((Q \land \neg R) \Rightarrow (\neg R U (P \land \neg R)))$

©2003 Keijo Heljanko, ©2004 Timo Latvala
The $LTL$ property patterns of the previous slides can be quite useful in practice.

There are also other patterns available expressing:

- Universality: "$P$ is true"—trivial to obtain from absence pattern
- Precedence: "$S$ precedes $P$"
- Response: "$S$ responds to $P$"
- Etc., etc.

These $LTL$ property patterns can be obtained through the Web page:

http://patterns.projects.cis.ksu.edu/

©2003 Keijo Heljanko, ©2004 Timo Latvala
Relating Boolean and Temporal Operators

\[
\begin{align*}
X(\varphi_1 \lor \varphi_2) & \equiv X\varphi_1 \lor X\varphi_2 \\
X(\varphi_1 \land \varphi_2) & \equiv X\varphi_1 \land X\varphi_2 \\
X\neg\varphi & \equiv \neg X\varphi \\
\Diamond(\varphi_1 \lor \varphi_2) & \equiv \Diamond\varphi_1 \lor \Diamond\varphi_2 \\
\neg\Diamond\varphi & \equiv \Box\neg\varphi \\
\Box(\varphi_1 \land \varphi_2) & \equiv \Box\varphi_1 \land \Box\varphi_2 \\
\neg\Box\varphi & \equiv \Diamond\neg\varphi \\
(\varphi_1 \land \varphi_2) \ U \psi & \equiv (\varphi_1 \ U \psi) \land (\varphi_2 \ U \psi) \\
\varphi \ U (\psi_1 \lor \psi_2) & \equiv (\varphi \ U \psi_1) \lor (\varphi \ U \psi_2)
\end{align*}
\]
Some Idempotence Laws

\[ \diamondsuit \diamondsuit \varphi \equiv \diamond \varphi \]
\[ \Box \Box \varphi \equiv \Box \varphi \]
\[ \diamondsuit \Box \diamondsuit \varphi \equiv \Box \diamondsuit \varphi \]
\[ \Box \Box \Box \varphi \equiv \diamond \Box \varphi \]
\[ \varphi U (\varphi U \psi) \equiv \varphi U \psi \]
Unfolding Laws

\[ \diamond \varphi \equiv \varphi \lor X\diamond \varphi \]

\[ \Box \varphi \equiv \varphi \land X\Box \varphi \]

\[ \varphi U \psi \equiv \psi \lor (\varphi \land X(\varphi U \psi)) \]

\[ \varphi R \psi \equiv \psi \land (\varphi \lor X(\varphi R \psi)) \]