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 π 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 π in M, such that $\pi_0 = s^0$ and $\pi \models \neg f$.*

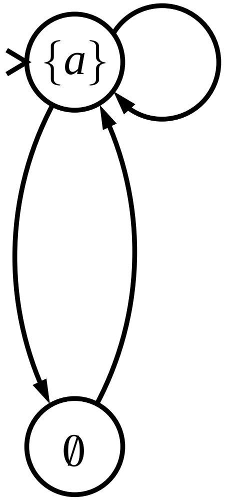
This definition is easier to handle, since to prove that f does not hold, we only have to find one path where $\neg f$ holds. If such a path cannot be found, then we can conclude that f holds.

©2003 Keijo Heljanko, ©2004 Timo Latvala

Universal Quantification

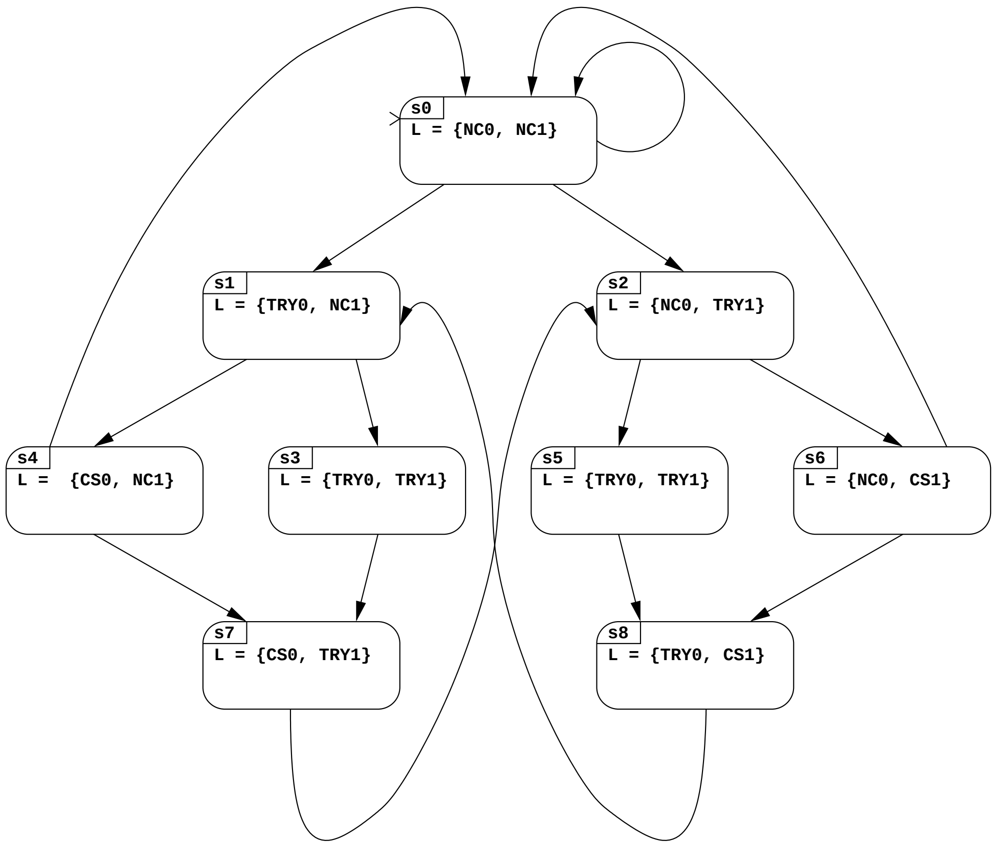
Note: It is possible that both $M \not\models f$ and $M \not\models \neg f$ hold.

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



ⓒ2003 Keijo Heljanko, ⓒ2004 Timo Latvala

Example: Properties of the Mutex System

- $M \models \Box\neg(\mathit{CR0} \wedge \mathit{CR1})$

  (both processes are never in their critical sections at the same time)

- $M \models \Box(\mathit{TRY0} \Rightarrow \Diamond \mathit{CR0})$

  (always when process 0 enters the trying section, it is eventually followed by process 0 entering the critical section)

- $M \models \Box(\mathit{TRY0} \Rightarrow (\mathit{TRY0}\, U\, \mathit{CR0}))$

  (always when process 0 enters the trying section, it stays in the trying section until it enters the critical section)

©2003 Keijo Heljanko, ©2004 Timo Latvala

Example: Unsatisfied Properties of the Mutex System

- $M \not\models \Diamond \mathit{CR0}$
  (the following does *not* hold: process 0 will eventually enter the critical section)

- $M \not\models \Box\Diamond \mathit{CR0} \Rightarrow \Box\Diamond \mathit{CR1}$
  (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)

©2003 Keijo Heljanko, ©2004 Timo Latvala

Example: More Properties of the Mutex System

- $M \models \Box\Diamond \mathit{CR0} \Rightarrow \Box\Diamond \mathit{TRY0}$
  (if process 0 is infinitely often in the critical section, then it is also infinitely often in the trying section)

- $M \models \Diamond\Box \mathit{NC0} \Rightarrow \Diamond\Box\neg \mathit{CR0}$
  (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)

- $M \models (\Box\Diamond \mathit{TRY0} \wedge \Box\Diamond \mathit{TRY1}) \Rightarrow (\Box\Diamond \mathit{CR0} \wedge \Box\Diamond \mathit{CR1})$
  (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)

©2003 Keijo Heljanko, ©2004 Timo Latvala

LTL Property Patters: 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

LTL Property Patters: Scope

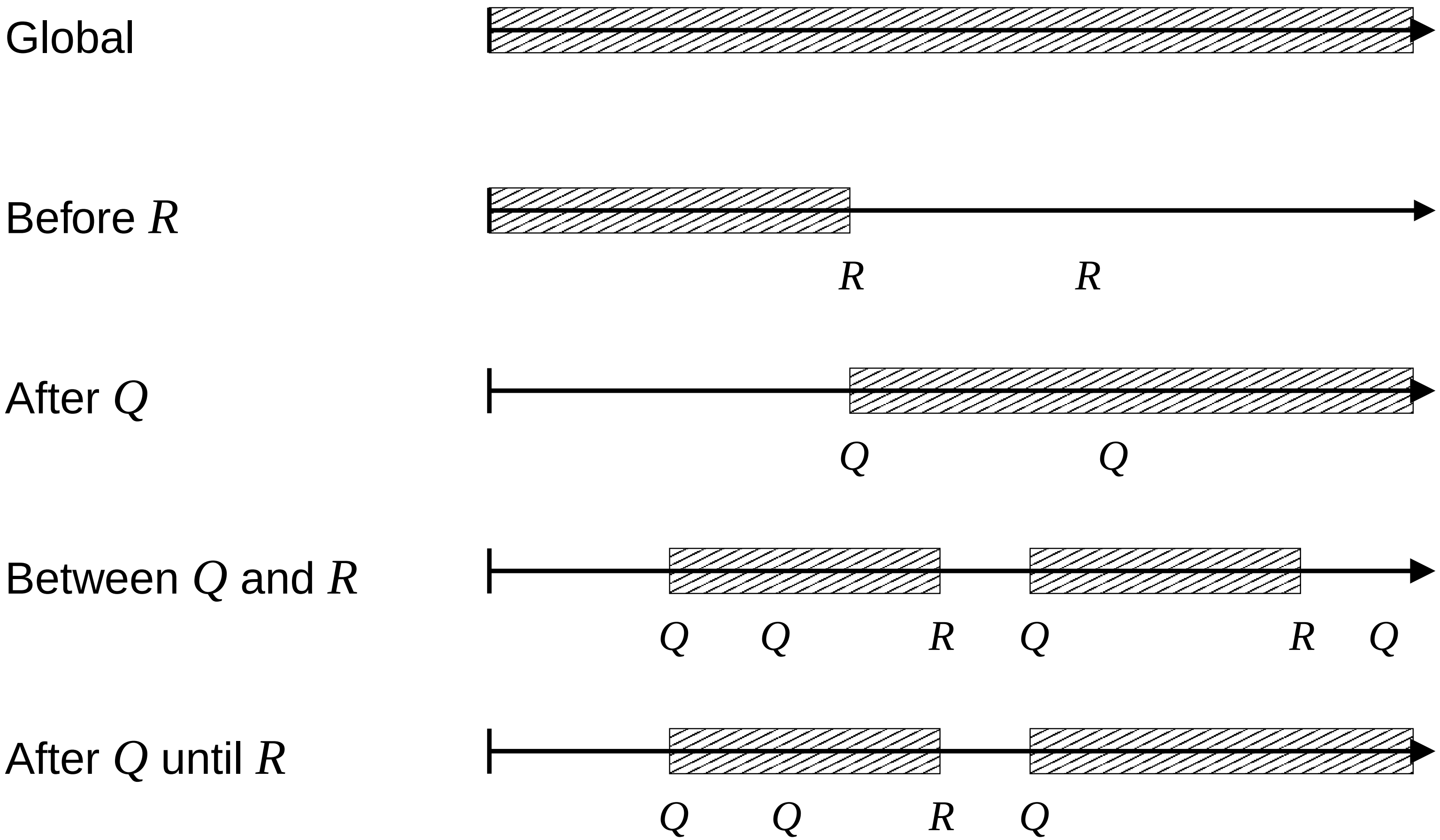
- *Between Q and R*: The scope of the requirement contains all sequences of indexes, such that Q holds at the first index until (but not including) the larger index where R holds at the first time.

- *After Q until R*: The scope of the requirement contains all sequences of indexes, such that:
  (i) Q holds at the first index until (but not including) the larger index where R holds at the first time, or
  (ii) Q holds at the first index but the R never holds at a larger index.

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 Patters: Scope



©2003 Keijo Heljanko, ©2004 Timo Latvala

LTL Property Patters: Absence

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: $\Box((Q \wedge \neg R \wedge \Diamond R) \Rightarrow (\neg P \; U \; R))$
- After Q until R: $\Box((Q \wedge \neg R) \Rightarrow (\neg P \; U \; (R \vee \Box(\neg P))))$

©2003 Keijo Heljanko, ©2004 Timo Latvala

LTL Property Patters: Existence

Existence pattern specifies that "P becomes true" within the scope:

- Global: $\Diamond P$
- Before R: $\neg R\, U\, ((P \wedge \neg R) \vee (\Box \neg R))$
- After Q: $(\Box(\neg Q)) \vee (\Diamond(Q \wedge (\Diamond P)))$
- Between Q and R: $\Box((Q \wedge \neg R) \Rightarrow (\neg R\, U\, ((P \wedge \neg R) \vee (\Box(\neg R)))))$
- After Q until R: $\Box((Q \wedge \neg R) \Rightarrow (\neg R\, U\, (P \wedge \neg R)))$

©2003 Keijo Heljanko, ©2004 Timo Latvala

LTL Property Patters

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

$$
\begin{array}{rcl}
X(\varphi_1 \vee \varphi_2) & \equiv & X\varphi_1 \vee X\varphi_2 \\
X(\varphi_1 \wedge \varphi_2) & \equiv & X\varphi_1 \wedge X\varphi_2 \\
X\neg\varphi & \equiv & \neg X\varphi \\[1ex]
\Diamond(\varphi_1 \vee \varphi_2) & \equiv & \Diamond\varphi_1 \vee \Diamond\varphi_2 \\
\neg\Diamond\varphi & \equiv & \Box\neg\varphi \\[1ex]
\Box(\varphi_1 \wedge \varphi_2) & \equiv & \Box\varphi_1 \wedge \Box\varphi_2 \\
\neg\Box\varphi & \equiv & \Diamond\neg\varphi \\[1ex]
(\varphi_1 \wedge \varphi_2)\, U\, \psi & \equiv & (\varphi_1\, U\, \psi) \wedge (\varphi_2\, U\, \psi) \\
\varphi\, U\, (\psi_1 \vee \psi_2) & \equiv & (\varphi\, U\, \psi_1) \vee (\varphi_1\, U\, \psi_2)
\end{array}
$$

©2003 Keijo Heljanko, ©2004 Timo Latvala

Some Idempotence Laws

$$
\begin{aligned}
\Diamond\Diamond\varphi &\equiv \Diamond\varphi \\
\Box\Box\varphi &\equiv \Box\varphi \\
\Diamond\Box\Diamond\varphi &\equiv \Box\Diamond\varphi \\
\Box\Diamond\Box\varphi &\equiv \Diamond\Box\varphi \\
\varphi\, U\, (\varphi\, U\, \psi) &\equiv \varphi\, U\, \psi
\end{aligned}
$$

©2003 Keijo Heljanko, ©2004 Timo Latvala

Unfolding Laws

$$
\begin{aligned}
\Diamond\varphi &\equiv \varphi \lor X \Diamond\varphi \\
\Box\varphi &\equiv \varphi \land X \Box\varphi \\
\varphi\, U\, \psi &\equiv \psi \lor (\varphi \land X(\varphi\, U\, \psi)) \\
\varphi\, R\, \psi &\equiv \psi \land (\varphi \lor X(\varphi\, R\, \psi))
\end{aligned}
$$

©2003 Keijo Heljanko, ©2004 Timo Latvala