# LTL - syntax and semantics

Franco Raimondi

Department of Computer Science
School of Science and Technology
Middlesex University
http://www.rmnd.net

# LTL syntax and semantics

# LTL Syntax (BNF)

$$\phi ::= \top \mid p \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc \phi \mid \phi U \phi$$

Derived operators:

- $\Diamond\phi \equiv \top U \phi$
- $\Box\phi \equiv \neg\Diamond\neg\phi$

I will use also: $X, F, G$ for $\bigcirc, \Diamond, \Box$.

# LTL Syntax: some examples

- Response: $G(\text{request} \implies (F\text{response}))$.
- Mutual exclusion: $G(\neg(c_1 \wedge c_2))$.
- Starvation freedom (from the book): $\bigwedge_i (GFw_i \rightarrow GFc_i)$.
- Traffic light (it is not red immediately after green): $G(\text{green} \implies \neg X\text{red})$.

(examples with U after semantics)

# LTL Semantics

It is given in terms of Transition Systems.

$$TS = (S, Act, \rightarrow, I, AP, L)$$

- A path $\pi$ is a sequence of states $s_0 s_1 \ldots$ s.t. $s_0 \in I$.

# LTL Semantics

It is given in terms of Transition Systems.

$$TS = (S, Act, \rightarrow, I, AP, L)$$

- A path $\pi$ is a sequence of states $s_0 s_1 \ldots$ s.t. $s_0 \in I$.
- The *trace* of a path $\pi$ is trace$(\pi) = L(s_0)L(s_1)\ldots$. Notice that this is a sequence of *sets of atomic propositions*

# LTL Semantics

It is given in terms of Transition Systems.

$$TS = (S, Act, \to, I, AP, L)$$

- A path $\pi$ is a sequence of states $s_0 s_1 \dots$ s.t. $s_0 \in I$.
- The *trace* of a path $\pi$ is $\text{trace}(\pi) = L(s_0)L(s_1)\dots$. Notice that this is a sequence of *sets of atomic propositions*
- If $\Pi$ is a set of paths, $\text{traces}(\Pi) = \{\mathrm{trace}(\pi) | \pi \in Pi\}$.

# LTL Semantics

It is given in terms of Transition Systems.

$$TS = (S, Act, \rightarrow, I, AP, L)$$

- A path $\pi$ is a sequence of states $s_0 s_1 \ldots$ s.t. $s_0 \in I$.
- The *trace* of a path $\pi$ is trace($\pi$) $= L(s_0)L(s_1)\ldots$. Notice that this is a sequence of *sets of atomic propositions*
- If $\Pi$ is a set of paths, traces($\Pi$) $= \{\mathrm{trace}(\pi) | \pi \in Pi\}$.
- Traces($s$) $=$ traces(Paths($s$))

# LTL Semantics

It is given in terms of Transition Systems.

$$TS = (S, Act, \rightarrow, I, AP, L)$$

- A path $\pi$ is a sequence of states $s_0 s_1 \ldots$ s.t. $s_0 \in I$.
- The *trace* of a path $\pi$ is $\text{trace}(\pi) = L(s_0)L(s_1)\ldots$. Notice that this is a sequence of *sets of atomic propositions*
- If $\Pi$ is a set of paths, $\text{traces}(\Pi) = \{\text{trace}(\pi) | \pi \in Pi\}$.
- $\text{Traces}(s) = \text{traces}(\text{Paths}(s))$
- $\text{Traces}(TS) = \bigcup_{s \in I} \text{Traces}(s)$.

# LTL Semantics

Let $\sigma$ be a word over $2^{AP}$. This means that $\sigma = A_0 A_1 \ldots$, where each $A_i$ is a set of atomic propositions.
Satisfaction is a relation $\models (2^{AP})^\omega \times \mathrm{LTL}$:

$$
\begin{array}{lll}
\sigma \models \top & & \\
\sigma \models p & \text{iff} & p \in A_0 \\
\sigma \models \phi \wedge \psi & \text{iff} & \sigma \models \phi \text{ and } \sigma \models \psi \\
\sigma \models X\phi & \text{iff} & \sigma[1:] \models \phi \\
\sigma \models \phi U \psi & \text{iff} & \exists i \geq 0 \text{ s.t. } \sigma[i:] \models \psi \\
 & & \text{and } \forall 0 \leq j < i, \sigma[j:] \models \phi
\end{array}
$$

[Examples with U]

# LTL Semantics

- $s \models \phi$ if Traces($s$) $\models \phi$ (i.e., each trace satisfies $\phi$).
- $TS \models \phi$ iff $s_0 \models \phi$ for all $s_0 \in I$.

# LTL Semantics: some equivalences

- $\neg X\phi \equiv X\neg\phi$
- $\neg F\phi \equiv G\neg\phi$ (and similarly for $\neg G\phi$)
- $FF\phi \equiv F\phi$
- $GG\phi \equiv G\phi$
- $F(\phi \vee \psi) \equiv F\phi \vee F\psi$
- $G(\phi \wedge \psi) \equiv G\phi \wedge G\psi$.

But

- $G(\phi \vee \psi) \not\equiv G\phi \vee G\psi$
- $F(\phi \wedge \psi) \not\equiv F\phi \wedge F\psi$.

**Show a TS for the last two formulae above**

# Weak Until Operator

Sometimes it is used in specifications.

- $\phi U \psi$ required $\psi$ to be true at some point in the future. It may be too strong:
- $\phi W \psi \equiv \phi U \psi \vee G \phi$
- Hence, $G\phi \equiv \phi W \bot$.

# Release

- $\phi R \psi$: intuitively, $\psi$ remains true if $\psi$ does not become true.
- Important equivalence: $\phi R \psi \equiv \neg(\neg\phi U \neg\psi)$
- $G\phi \equiv \bot R \phi$ (and $\equiv \phi W \bot$, see above)

This equivalence is important to convert LTL formulae to positive normal form (PNF). Intuitively, "all negations pushed in":

- $\neg(\phi \wedge \psi) \equiv \neg\phi \wedge \neg\psi$
- $\neg X\phi \equiv X\neg\phi$
- $\neg(\phi U \psi) \equiv \neg\phi R \neg\psi$

We will use these tomorrow to build a Büchi automaton that accepts the same words of any given LTL formula.