1 Formulas of LTL in Plain English

- Some useful formulas of LTL and their translations into plain English:

1. \[ \psi = G(\varphi \rightarrow X \neg \varphi) \]
   - "Globally, if \( \varphi \) holds, then \( X \neg \varphi \) holds."
   - Or: "If at any point \( \varphi \) is true, then \( X \neg \varphi \) is also true at that point."
   - So, if at any point on some path \( \pi \), we enter a state \( s_i \) where \( \varphi \) is true, then it must be true that \( \varphi \) is false for state \( s_{i+1} \).
   - But notice that if for some path \( \pi \), \( \varphi \) is never true, then \( \pi \models \psi \) as well.

2. \[ \psi = G(\varphi \rightarrow G \varphi) \]
   - "If \( \varphi \) is true at any point, then \( \varphi \) will be true for every state after that point as well."
   - Again, if for some path \( \pi \), \( \varphi \) is never true, then \( \pi \models \psi \).

3. \[ \psi = G(\varphi \rightarrow F \varphi) \]
   - "Globally, once \( \varphi \) is true for some state, then there must exist some state in the future for which \( \varphi \) is also true."
   - For a finite path, \( \psi \) means that \( \varphi \) will be true at least twice. For an infinite path, \( \psi \) means that \( \varphi \) will be true infinitely many times, since every time you encounter a state for which \( \varphi \) is true, then it must be the case that \( \varphi \) is true again sometime in the future.

4. \[ \psi = G(\varphi \rightarrow XG \neg \varphi) \]
   - "Globally, once \( \varphi \) is true for some state, then from the next state on (forever), \( \varphi \) will be false."
   - Or: "\( \varphi \) is true at most once, possibly never."

5. \[ \psi = F(\varphi \land XF \varphi) \]
   - "\( \varphi \) is true in at least two states."
   - \( F(\varphi \land F \varphi) \) wouldn’t mean the same thing, because \( F \) includes the present. So, we need \( X \) to precede it.

6. \[ \psi = GF \varphi \]
   - "\( \varphi \) holds infinitely often."
   - Or: "Globally, you will always encounter another state for which \( \varphi \) is true."

7. \[ \psi = FG \varphi \]
• "\(\varphi\) is eventually always true."
• Or: "There exists some future state after which \(\varphi\) is always true."

8. \(\psi = (X\varphi \rightarrow \varphi)\)
• "If \(\varphi\) is true in the next state, then \(\varphi\) is true in the preceding state as well."

9. \(\psi = (X\varphi \rightarrow \varphi) \land G(XX\varphi \rightarrow \varphi \lor X\varphi)\)
• "Globally, if \(\varphi\) is true in the state after the next (two states in the future), then either \(\varphi\) is true in the present state, or \(\varphi\) is true in the next state."

2 Deadlocks
• LCS doesn’t provide formulas for expressing deadlocks and finite paths, so some useful ones will be presented here.

1. \(\psi = X\bot\)
   • "There is no next state."

2. \(\psi = G(X\bot \rightarrow \text{terminal})\)
   • "Globally, if there is no next state from the present state, then the present state is terminal."
   • Or: "The system is free of deadlocks."

3. \(\psi = F(X\bot \land \neg \text{terminal})\)
   • "A deadlock state can be reached"
   • Negation of the formula immediately above it.

4. \(\psi = FX\bot\)
   • "Every execution path is finite."

5. \(\psi = GX\top\)
   • "There will always be a next state."
   • Or: "Every execution path is infinite."

3 Inexpressible Statements in LTL
1. "\(\varphi\) is true in every odd state and false in every even state."
   • Can’t express it in LTL, try it.

2. As an exercise, try explaining the following in English:
   • \(G(\varphi \rightarrow XX\varphi)\)
   • \(\varphi \land G(\varphi \rightarrow XX\varphi)\)
4 Expressing Responsiveness of a System in LTL

1. "Every request is eventually acknowledged."
   - $\text{G} \ (\text{request} \rightarrow \text{XF acknowledged})$
     - Globally, once we reach a state where "request" is true, then beginning at the next state, it will eventually be true that "acknowledged" will be true.

2. "Every request remains true until it is acknowledged."
   - $\text{G} \ (\text{request} \rightarrow (\text{request} \ U \ \text{acknowledged}))$
     - Globally, once we reach a state where "request" is true, then "request" will remain true until "acknowledged is true.

3. "Every request remains true until it is acknowledged, after which it immediately becomes false."
   - $\text{G} \ (\text{request} \rightarrow (\text{request} \land \neg \text{acknowledged}) \ U (\neg \text{request} \land \text{acknowledged}))$

4. Look up SPIN, a model checker for LTL

5 Translating Propositional LTL into FOL

1. Consider FOL models $M$ over $\mathbb{N}$ with $F = \emptyset$ and $R = (<) \ U$ (Propositional Variables used as Unary Predicates)
   - Sometimes called "First Order Monadic Logic of Linear Order"
   - Page 2 of Handout 20 lists the translation functions from LTL into FOL

2. **Theorem**: Let $M$ be a transition system, i.e. - $M$ is an LTL model, and $\pi$ is a path in $M$. The following are equivalent assertions, for every WFF $\varphi$ of LTL and every $i \leq 0$:
   - $\pi^i \models_{\text{LTL}} \varphi$
   - There is an FOL model $N$ such that $N \models_{\text{FOL}} [\varphi] (i)$

3. **Now**: Is there a translation from FOL into LTL? That is, is it possible to define a WFF $\psi$ of LTL for every WFF $\varphi$ of FOL such that $\varphi$ is semantically valid iff $\psi$ is semantically valid?
   - **Answer**: Yes, by Kamp’s Theorem.