

Linear Temporal Logic

1. Overview

Linear Temporal Logic (LTL) is a branch of temporal logic, used to reason about sequences of events over time. It abstracts time into a sequence of states, where each state has a unique successor (hence the linearity).

2. Practical Applications

LTL is used to specify and verify the behavior of systems, especially focusing on properties like safety and liveness.

3. Atomic Propositions

An atomic proposition p is a boolean expression or statement that can be TRUE or FALSE (e.g., p = "it is raining"). Atomic propositions can be simple (e.g., $x < 8$), but they can also be more complex temporal logic formulas, consisting of temporal operators.

4. Logical Operators

- Negation (NOT)
 - Let p = "it is raining", so $\neg p$ = "it is not raining"
- Conjunction (AND)
 - Let p = "it is raining", q = "I have an umbrella", so $p \wedge q$ = "it is raining AND I have an umbrella"
- Disjunction (OR)
 - Let p = "it is raining", q = "I have an umbrella", so $p \vee q$ = "it is raining OR I have an umbrella"
- Implies (IF ..., THEN ...)
 - Let p = "there is an error", q = "I will resolve the error", so $(p \rightarrow q)$ = "IF there is an error, THEN I will resolve the error"

5. Temporal Operators

- F – Finally: This property exists in at least 1 state
- G – Globally: This property holds in all states
- X – Next: This property holds in the next state
- W – Weak until: Property A is TRUE at least until property B becomes TRUE, or property A stays TRUE forever
- R – Release: Property A is TRUE exactly until property B becomes TRUE, or property A stays TRUE forever

6. Putting It All Together: Example

- Distributed system
 - Definitions (Atomic Propositions)
 - *process1critical*: process 1 is in the critical section
 - *process2critical*: process 2 is in the critical section
 - *process1wants*: process 1 wishes to move into the critical section
 - Constraints
 - Mutual exclusion: no more than 1 process is in a given critical section at any given time
 - $G \neg (process1critical \wedge process2critical)$
 - No starvation: if a process wishes to move into a critical section, then it will do so at some point
 - $G (process1wants \rightarrow F process1critical)$