Linear Temporal Logic - Syntax

\[ \phi ::= p \mid (\phi) \mid \neg \phi \mid \phi \land \phi' \mid \phi \lor \phi' \mid \circ \phi \mid \phi U \phi' \mid \phi V \phi' \mid \Box \phi \mid \Diamond \phi \]

- \( p \) – a proposition over state variables
- \( \circ \phi \) – “next”
- \( \phi U \phi' \) – “until”
- \( \phi V \phi' \) – “releases”
- \( \Box \phi \) – “box”, “always”, “forever”
- \( \Diamond \phi \) – “diamond”, “eventually”, “sometime”
LTL Semantics: The Idea

\[
p \rightarrow p
\]

\[
\circ \varphi \rightarrow \varphi
\]

\[
\varphi U \psi \rightarrow \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \psi
\]

\[
\varphi V \psi \rightarrow \psi, \psi, \psi, \psi, \psi, \psi, \varphi, \psi
\]

\[
\square \varphi \rightarrow \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi, \varphi
\]

\[
\Diamond \varphi \rightarrow \varphi
\]
Formal LTL Semantics

Given:

- $G = (V, F, af, R, ar)$ signature expressing state propositions
- $Q$ set of states,
- $M$ modeling function over $Q$ and $G$: $M(q, p)$ is true iff $q$ models $p$. Write $q \models p$.
- $\sigma = q_0q_1\ldots q_n\ldots$ infinite sequence of state from $Q$.
- $\sigma^i = q_iq_{i+1}\ldots q_n\ldots$ the $i^{th}$ tail of $\sigma$

Say $\sigma$ models LTL formula $\varphi$, write $\sigma \models \varphi$ as follows:

- $\sigma \models p$ iff $q_0 \models p$
- $\sigma \models \neg \varphi$ iff $\sigma \not\models \varphi$
- $\sigma \models \varphi \land \psi$ iff $\sigma \models \varphi$ and $\sigma \models \psi$.
- $\sigma \models \varphi \lor \psi$ iff $\sigma \models \varphi$ or $\sigma \models \psi$. 

Formal LTL Semantics

- \( \sigma \models \Diamond \varphi \) iff for some \( i \), \( \sigma^i \models \varphi \)
- \( \sigma \models \varphi \) iff for some \( k \), \( \sigma^k \models \psi \) and for all \( i < k \), \( \sigma^i \models \varphi \)
- \( \sigma \models \varphi \) iff for some \( k \), \( \sigma^k \models \varphi \) and for all \( i \leq k \), \( \sigma^i \models \psi \), or for all \( i \), \( \sigma^i \models \psi \).
- \( \sigma \models \Box \varphi \) if for all \( i \), \( \sigma^i \models \psi \)
- \( \sigma \models \lozenge \varphi \) if for some \( i \), \( \sigma^i \models \psi \)
Some Common Combinations

- $\Box \Diamond p$ “$p$ will hold infinitely often”
- $\Diamond \Box p$ “$p$ will continuously hold from some point on”
- $(\Box p) \Rightarrow (\Box q)$ “if $p$ happens infinitely often, then so does $q$"
Some Equivalences

- \( \Box(\varphi \land \psi) = (\Box \varphi) \land (\Box \psi) \)
- \( \Diamond(\varphi \lor \psi) = (\Diamond \varphi) \lor (\Diamond \psi) \)
- \( \Box \varphi = F \lor \varphi \)
- \( \Diamond \varphi = T \lor \varphi \)
- \( \varphi \lor \psi = \neg ((\neg \varphi) \lor (\neg \psi)) \)
- \( \varphi \lor \psi = \neg ((\neg \varphi) \lor (\neg \psi)) \)
- \( \neg (\Diamond \varphi) = \Box (\neg \varphi) \)
- \( \neg (\Box \varphi) = \Diamond (\neg \varphi) \)
Some More Equivalences

- □φ = φ \land \circ □φ
- ◊φ = φ \lor \circ ◊φ
- φ \lor ψ = (φ \land ψ) \lor (ψ \land \circ (φ \lor ψ))
- φ U ψ = ψ \lor (φ \land \circ (φ \lor ψ))

- □, ◊, U, V may all be understood recursively, by what they state about right now, and what they state about the future
- Caution: □ vs ◊, U vs V differ in there limit behavior
Traffic Light Example

Basic Behavior:

- $\Box((\text{NSC} = \text{Red}) \lor (\text{NSC} = \text{Green}) \lor (\text{NSC} = \text{Yellow}))$  
- $\Box((\text{NSC} = \text{Red}) \Rightarrow ((\text{NSC} \neq \text{Green}) \land (\text{NSC} \neq \text{Yellow}))$  
- Similarly for $\text{Green}$ and $\text{Red}$  
- $\Box(((\text{NCS} = \text{Red}) \land \circ(\text{NCS} \neq \text{Red})) \Rightarrow \circ(\text{NCS} = \text{Green}))$  
- Same as $\Box((\text{NCS} = \text{Red}) \Rightarrow ((\text{NCS} = \text{Red}) \cup (\text{NCS} = \text{Green})))$  
- $\Box(((\text{NCS} = \text{Green}) \land \circ(\text{NCS} \neq \text{Green})) \Rightarrow \circ(\text{NCS} = \text{Yellow}))$  
- $\Box(((\text{NCS} = \text{Yellow}) \land \circ(\text{NCS} \neq \text{Yellow})) \Rightarrow \circ(\text{NCS} = \text{Red}))$  
- Same for $\text{EWC}$
Traffic Light Example

Basic Safety

\[ \Box((NSC = Red) \lor (EWC = Red)) \]
\[ \Box((NSC = Red) \land (EWC = Red)) \lor ((NSC \neq Green) \Rightarrow (\Diamond(NSC = Green)))) \]

Basic Liveness

\[ (\Diamond(NSC = Red)) \land (\Diamond(NSC = Green)) \land (\Diamond(NSC = Yellow)) \]
\[ (\Diamond(EWC = Red)) \land (\Diamond(EWC = Green)) \land (\Diamond(EWC = Yellow)) \]
Proof System for LTL

- First step: View $\varphi \bigvee \psi$ as macro: $\varphi \bigvee \psi = \neg((\neg \varphi) U (\neg \psi))$
- Second Step: Extend all rules of Prop Logic to LTL
- Third Step: Add one more rule: $\text{Gen} \quad \varphi$
- Fourth Step: Add a collection of axioms (a sufficient set of 8 exists)
  - A1: $\Box \varphi \iff \neg(\Diamond (\neg \varphi))$
  - A2: $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box \varphi \Rightarrow \Box \psi)$
  - A3: $\Box \varphi \Rightarrow (\varphi \land \circ \Box \varphi)$
  - A4: $\circ \neg \varphi \iff \neg \circ \varphi$
  - A5: $\circ(\varphi \Rightarrow \psi) \Rightarrow (\circ \varphi \Rightarrow \circ \psi)$
  - A6: $\Box(\varphi \Rightarrow \circ \varphi) \Rightarrow (\varphi \Rightarrow \Box \varphi)$
  - A7: $\varphi U \psi \iff (\varphi \land \psi) \lor (\varphi \land \circ (\varphi \bigvee \psi))$
  - A8: $\varphi U \psi \Rightarrow \Diamond \psi$

- Result: a sound and relatively complete proof system
- Can implement in Isabelle in much the same way as we did Hoare Logic
Important Meta-Definitions

- **A** is **sound** with respect to **B** if things that are “true” according to **A** are things that are “true” according to **B**.
- **A** is **complete** with respect to **B** if things that are “true” according to **B** are things that are “true” according to **A**.
- **A** is **sound** if things that are “true” according to **A** are true.
- **A** is **complete** everything that is true (that is in the scope of **A**) is “true” according to **A**.
- **A** is **relatively complete** with respect to **B** if **A** is complete when **B** is.

Think: **A** proof system, **B** mathematical model; or **A** a proof system, **B** a subsystem.
Exercise: \( \varphi \land \psi \Rightarrow \Box \varphi \land \Box \psi \)
What is Model Checking?

Most generally **Model Checking** is

- an **automated** technique, that given
- a **finite-state** model $M$ of a system
- and a **logical** property $\varphi$,
- **checks** whether the property holds of model: $M \models \varphi$?
Model Checking

- Model checkers usually give example of failure if $M \not\models \varphi$.
- This makes them useful for debugging.
- **Problem:** Can only handle finite models: unbounded or continuous data sets can’t be directly handled
  - Symbolic model checking can handle limited cases of finitely presented models
- **Problem:** Number of states grows exponentially in the size of the system.
- **Answer:** Use abstract model of system
- **Problem:** Relationship of results on abstract model to real system?
LTL Model Checking

- **Model Checking Problem:** Given model $M$ and logical property $\varphi$ of $M$, does $M \models \varphi$?

- Given transition system with states $Q$, transition relation $\delta$ and initial state state $I$, say $(Q, \delta, I) \models \varphi$ for LTL formula $\varphi$ if every run of $(Q, \delta, I)$, $\sigma$ satisfies $\sigma \models \varphi$.

**Theorem**

*The Model Checking Problem for finite transition systems and LTL formulae is decideable.*

- Treat states $q \in Q$ as letters in an alphabet.
- Language of $(Q, \delta, I)$, $L(Q, \delta, I)$ (or $L(Q)$ for short) is set of runs in $Q$.
- Language of $\varphi$, $L\varphi = \{\sigma | \sigma \models \varphi\}$
- Question: $L(Q) \subseteq L(\varphi)$?
- Same as: $L(Q) \cap L(\neg \varphi) = \emptyset$?
How to Decide the Model Checking Problem?

- How to answer $\mathcal{L}(Q) \cap \mathcal{L}(\neg \varphi) = \emptyset$?
- Common approach:
  - Build automaton $A$ such the $\mathcal{L}(A) = \mathcal{L}(Q) \cap \mathcal{L}(\neg \varphi)$
  - Are accepting states of $A$ reachable? (Infinitely often?)
- How to build $A$?
  - One possible answer: Build a series of automata by recursion on structure of $\neg \varphi$.
  - Another possible answer: Build an automaton $B$ such $\mathcal{L}(B) = \mathcal{L}(\neg \varphi)$; take $A = B \times Q$
- Will do at least one approach if time after Spin