Example: Traffic Lights

Example (cont)

- LTS for traffic light has $3 \times 3 \times 2 = 18$ possible well typed states
  - Is it possible to reach a state where $\text{NSC} \neq \text{Red} \land \text{EWC} \neq \text{Red}$ from an initial state?
  - If so, what sequence of actions allows this?
  - Do all the immediate predecessors of a state where $\text{NSC} \land \text{EWC} = \text{Green}$ satisfy $\text{NSC} = \text{Red} \land \text{EWC} = \text{Red}$?
  - If not, are any of those offended states reachable from the initial state, and if so, how?

- LTS for Mutual Exclusion has $6 \times 6 \times 2 = 144$ possible well typed states.
  - Is it possible to reach a state where $\text{pc1} = m5 \land \text{pc2} = n5$?
  - How can we state these questions rigorously, formally?
  - Can we find an algorithm to answer these types of questions?

Linear Temporal Logic - Syntax

$LTL$ Semantics: The Idea

- $\phi$ - a proposition over state variables
- $\phi'$ - "next"
- $\phi \lor \phi'$ - "until"
- $\phi \land \phi'$ - "releases"
- $\phi$ - "box", "always", "forever"
- $\Diamond \phi$ - "diamond", "eventually", "sometime"
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 $q \models p$.
- $\sigma = q_0 q_1 \ldots q_n \ldots$ infinite sequence of state from $Q$.
- $\sigma^i = q_i q_{i+1} \ldots q_n \ldots$ the $i^{th}$ tail of $\sigma$.

Say $\sigma$ models LTL formula $\varphi$, write $\sigma \models \varphi$ as follows:
- $\sigma \models p$ iff $q_0 \models p$.
- $\sigma \models \neg p$ if $\sigma \models \varphi$.
- $\sigma \models \varphi \land \psi$ if $\sigma \models \varphi$ and $\sigma \models \psi$.
- $\sigma \models \varphi \lor \psi$ if $\sigma \models \varphi$ or $\sigma \models \psi$.
- $\sigma \models \square \varphi$ if $\sigma \models \varphi$ for all $i$.
- $\sigma \models \diamond \varphi$ if for some $i$, $\sigma^i \models \varphi$.

Some Common Combinations

- $\square p$ “$p$ will hold infinitely often”
- $\diamond p$ “$p$ will continuously hold from some point on”
- $(\square p) \Rightarrow (\diamond q)$ “if $p$ happens infinitely often, then so does $q$.

Some More Equivalences

- $\square \varphi = \varphi \land \diamond \varphi$
- $\diamond \varphi = \varphi \lor \diamond \varphi$
- $\varphi \lor \psi = (\varphi \land \psi) \lor (\varphi \land \diamond \psi)$
- $\varphi \lor \psi = (\varphi \lor \psi) \lor (\varphi \land \diamond \psi)$
- $\square, \diamond, U, V$ may all be understood recursively, by what they state about right now, and what they state about the future.
- Caution: $\square$ vs $\diamond$, $U$ vs $V$ differ in there limit behavior.

Traffic Light Example

Basic Behavior:
- $\square ((\text{NC} = \text{Red}) \lor (\text{NC} = \text{Green}) \lor (\text{NC} = \text{Yellow}))$
- $\square ((\text{NC} = \text{Red}) \Rightarrow ((\text{NC} = \text{Green}) \land (\text{NC} = \text{Yellow}))$
- Similarly for Green and Red
- $\square ((\text{NC} = \text{Red}) \land (\text{NC} \neq \text{Red}) \Rightarrow (\text{NC} = \text{Green}))$
- Same as $\square ((\text{NC} = \text{Red}) \Rightarrow ((\text{NC} = \text{Red}) \land (\text{NC} = \text{Green})))$
- $\square ((\text{NC} = \text{Green}) \land (\text{NC} \neq \text{Green}) \Rightarrow (\text{NC} = \text{Yellow}))$
- $\square ((\text{NC} = \text{Yellow}) \land (\text{NC} \neq \text{Yellow}) \Rightarrow (\text{NC} = \text{Red}))$
- Same for EWC.
Traffic Light Example

Basic Safety
- \(\square((\text{NSC} = \text{Red}) \lor (\text{EWC} = \text{Red}))\)
- \(\square(((\text{NSC} = \text{Red}) \land (\text{EWC} = \text{Red})) \lor ((\text{NSC} \neq \text{Green}) \Rightarrow (\diamond(\text{NSC} = \text{Green}))))\)

Basic Liveness
- \((\diamond(\text{NSC} = \text{Red})) \land (\diamond(\text{NSC} = \text{Green})) \land (\diamond(\text{NSC} = \text{Yellow})))\)
- \((\diamond(\text{EWC} = \text{Red})) \land (\diamond(\text{EWC} = \text{Green})) \land (\diamond(\text{EWC} = \text{Yellow})))\)

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 if 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.

Proof System for LTL
- First step: View \(\varphi \lor \psi\) as macro: \(\varphi \lor \psi = \neg((\neg \varphi) U(\neg \psi))\)
- Second Step: Extend all rules of Prop Logic to LTL
  - \(\Box \varphi\)
- Third Step: Add one more rule: \(\Box \varphi\)
- Fourth Step: Add a collection of axioms (a sufficient set of 8 exists)
  - A1: \(\Box \varphi \Rightarrow \neg(\Box(\neg \varphi))\)
  - A2: \(\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box \varphi \Rightarrow \Box \psi)\)
  - A3: \(\Box \varphi \Rightarrow (\varphi \Rightarrow \Box \varphi)\)
  - A4: \(\neg \varphi \Rightarrow \neg \Box \varphi\)
  - A5: \(\Box(\varphi \Rightarrow \psi) \Rightarrow (\neg \psi \Rightarrow \Box \varphi)\)
  - A6: \(\Box(\varphi \Rightarrow \Box \psi) \Rightarrow (\varphi \Rightarrow \Box \psi)\)
  - A7: \(\psi U \varphi \Rightarrow (\varphi \land \psi) \lor (\varphi \land o(\varphi V \psi))\)
  - A8: \(\varphi U \psi \Rightarrow \Box \psi\)
- Result: a sound and relatively complete proof system
- Can implement in Isabelle in much the same way as we did Hoare Logic

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
- 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 decidable.

- 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: $L\varphi \subseteq L$?
- Same as: $L(Q) \cap L(\neg \varphi) = \emptyset$