the two traffic lights should never show a green light at the same time

something bad never happens
traffic lights showing red should eventually show green

something good eventually happens
traffic lights showing red should eventually show green

something good eventually happens
traffic lights showing red should eventually show green

something good eventually happens
• classical propositional temporal logic (PTL)
  – first extensive use of a temporal logic in computer science [Pnueli]

• semantics with respect to paths / traces / state-sequences of a system
  – focus on single runs of a system

• LTL features in comparison to other temporal logics:
  – easy to understand (actually a matter of taste)
  – compositional (supports modular proofs)
  – somewhat expensive to check
Examples of basic LTL Formulae

$\mathcal{X} p$  
next time $p$ holds
immediately after the current state $p$ holds

$\neg p$

$p$

0

$\mathcal{F} p$

basic liveness property: eventually (finally) $p$ holds
after finite number of steps $p$ holds

$p$

$p$

0

Formal Verification #251-0247-00 – WS 2006/2007 – Daniel Kröning – ETH Zürich
Examples of basic LTL Formulae

$Gp$ basic safety property: $p$ holds globally
after any number of steps $p$ holds

$p$

$¬p$

$0$

$p U q$ $p$ holds until $q$ holds
after a finite number of steps $q$ holds and
on the way to this point $p$ continuously holds

$p$

$¬p$

$q$

$¬q$
\( p \mathbf{R} q \) \( p \) releases \( q \) (if ever \( q \) stops to hold)

after a finite number of steps \( p \) holds and

on the way to this point, including it, \( q \) continuously holds

or \( q \) continuously holds all the time
• the set of atoms $A$ consist of propositional variables $p, q, \ldots$

• boolean operators $\land, \lor, \neg, \rightarrow, \ldots$

• unary temporal operators (we prefer textual version)
  
  – *next* time operator $\mathbf{X}$ ($\bigcirc$ in PTL)
  
  – *finally* operator $\mathbf{F}$ ($\Diamond$ in PTL)
  
  – *globally* operator $\mathbf{G}$ ($\square$ in PTL)

• binary *until* operator $\mathbf{U}$ (strong version)
  
  – we also use its dual $\mathbf{R}$, the *release* operator
• temporal logic formulae are interpreted over *Kripke Structures*
  
  – some variants allow *Labelled Transition Systems* (LTS)

• a Kripke structure \( K = (S, T, I, L) \) consists of
  
  – a set of states \( S \)
  
  – a total transition relation \( T \subseteq S \times S \)
  
  – a non empty set of initial states \( I \subseteq S \)
  
  – a labelling of states with atoms \( L: S \rightarrow 2^A \)

• a transition relation is called **total** iff \( \forall s \in S \, \exists t \in S \left[ (s, t) \in T \right] \)
• Kripke structures have their origin in modal logics

  – generalization of temporal logics

• the propositional variables are the observables

• a labelling of a state can be interpreted as an observation

• or as an assignment to the propositional variables

  \[ \sigma_s(p) = 1 \iff p \in L(s) \]

• transition relation (indirectly) relates observations or propositional assignments
• we say a state $s$ has a transition to a state $t$ (written $s \rightarrow t$) iff $(s, t) \in T$

• a path $\pi$ is an infinite sequence of states $\pi = (s_0, s_1, \ldots) \in S^\omega$ with

$\quad s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$

• we define $\pi(i) \equiv s_i$ and $\pi^i \equiv (s_i, s_{i+1}, \ldots)$

• Note: infinite assumption about paths, totality of transition relation and non-emptiness of initial states set depend on each other and are only necessary to simplify semantics.
we recursively define \( f \) to be valid on path \( \pi \), written \( \pi \models f \), as follows:

\[
\begin{align*}
\pi \models p & \quad \text{iff} \quad p \in L(\pi(0)) & \quad \text{(first state of } \pi \text{ is labelled with } p) \\
\pi \models \neg g & \quad \text{iff} \quad \pi \nmid g & \quad \text{(} g \text{ is not valid on } \pi) \\
\pi \models g \land h & \quad \text{iff} \quad \pi \models g \text{ and } \pi \models h & \quad \text{(} g \text{ and } h \text{ are both valid on } \pi) \\
\pi \models g \lor h & \quad \text{iff} \quad \pi \models g \text{ or } \pi \models h & \quad \text{(one of } g \text{ or } h \text{ is valid on } \pi) \\
\pi \models g \rightarrow h & \quad \text{iff} \quad \text{if } \pi \models g \text{ then } \pi \models h & \quad \text{(if } g \text{ is valid on } \pi \text{ then } h \text{ too)}
\end{align*}
\]

(semantics for further operators can be defined accordingly)
\[ \pi \models Xg \quad \text{iff} \quad \pi^1 \models g \quad (g \text{ is valid on } \pi \text{ with first state chopped off}) \]

\[ \pi \models Fg \quad \text{iff} \quad \exists i[\pi^i \models g] \quad (g \text{ is valid on some suffix of } \pi) \]

\[ \pi \models Gg \quad \text{iff} \quad \forall i[\pi^i \models g] \quad (g \text{ is valid on every suffix of } \pi) \]

\[ \pi \models g \bigcup h \quad \text{iff} \quad \exists i[\pi^i \models h \text{ and } \forall j < i[\pi^j \models g]] \]

\[ \pi \models g \bigcap h \quad \text{iff} \quad \pi \models \neg(\neg g \bigcup \neg h) \]

\[ \quad \text{iff} \quad \forall i[\pi^i \models h \text{ or } \exists j < i[\pi^j \models g]] \]

\[ \quad \text{iff} \quad \forall i[\pi^i \models h] \text{ or } \exists j[\pi^j \models g \text{ and } \forall i \leq j[\pi^i \models h]] \]
mutual exclusion

$\text{at most one request is acknowledged}$

$\text{it is infinitely often my turn}$

$(\text{myTurn will always eventually hold})$

entering the $\text{try}$ region will eventually . . .

. . . lead to the $\text{critical}$ region

also known as $\text{request-acknowledge pattern}$

entering the $\text{try}$ region . . .

. . . $\text{try}$ holds as long $\text{critical}$ does not hold

($\text{. . . try}$ is not released if $\text{critical}$ does not hold)

after initialization the systems stays initialized

$G(\neg (\text{critical}_1 \land \text{critical}_2))$

$\land_{i < j} G(\neg (\text{ack}_i \land \text{ack}_j))$

$GF \text{ myTurn}$

$G(\text{try} \rightarrow F \text{ critical})$

$G(\text{req} \rightarrow F \text{ ack})$

$G(\text{try} \rightarrow (\text{critical} R \text{ try}))$

($F \text{ try}$ is not released if $\text{critical}$ does not hold)

$FG \text{ initialized}$
Example

[Katoen02]

Determine which formulas are true.

1. $XXy$
2. $XXr$
3. $Fy$
4. $GFy$
5. $Fg$
6. $GFg$
7. $G\neg b$
8. $F\neg b$
9. $\neg y \cup y$
10. $\neg r \cup g$
11. $\neg b \cup b$
12. $b \cup \neg b$
Determine which formulas are true.

\[
\begin{align*}
1. & \quad \forall \quad \exists \\
& \quad XXy \quad \{1\} \quad \{1,2\} \quad \{1,2,3\} \\
2. & \quad XXr \quad \{\} \quad \{3,4\} \\
3. & \quad Fy \quad \{1,2,3,4\} \quad \{1,2,3,4\} \\
4. & \quad GFy \quad \{1,2,3,4\} \quad \{1,2,3,4\} \\
5. & \quad Fg \quad \{1,3\} \quad \{1,2,3,4\} \\
6. & \quad GFg \quad \{\} \quad \{1,2,3,4\} \\
7. & \quad G\neg b \quad \{\} \quad \{1,2,3\} \\
8. & \quad F\neg b \quad \{1,2,3,4\} \quad \{1,2,3,4\} \\
9. & \quad \neg y \cup y \quad \{1,2,3,4\} \quad \{1,2,3,4\} \\
10. & \quad \neg r \cup g \quad \{3\} \quad \{3\} \\
11. & \quad \neg b \cup b \quad \{4\} \quad \{1,2,3,4\} \\
12. & \quad b \cup \neg b \quad \{1,2,3,4\} \quad \{1,2,3,4\}
\end{align*}
\]
Between an elevator is called at a floor and the time it opens its doors at that floor, the elevator can arrive at that floor at most twice.

\[
\begin{align*}
\mathbf{G}((\text{call} \land \mathbf{F} \text{ open}) \rightarrow & \\
((\neg \text{atfloor} \land \neg \text{open}) \mathbf{U} & \\
(\text{open} \lor ((\text{atfloor} \land \neg \text{open}) \mathbf{U} & \\
(\text{open} \lor ((\neg \text{atfloor} \land \neg \text{open}) \mathbf{U} & \\
(\text{open} \lor ((\text{atfloor} \land \neg \text{open}) \mathbf{U} & \\
(\text{open} \lor ((\neg \text{atfloor} \mathbf{U} \text{ open}))))))))
\end{align*}
\]