Program correctness
Linear Time Temporal Logic

Marcello Bonsangue
Formal Verification

Verification techniques comprise

- a modelling framework $M$
  to describe a system

- a specification language $\phi$
  to describe the properties to be verified

- a verification method $M \models \phi$, $\Gamma \vdash \phi$
  to establish whether a model satisfies a property
Motivations

- For an elevator system, consider the requirements:
  - any request must ultimately be satisfied
  - the elevator never traverses a floor for which a request is pending without satisfying it

- Both concern the **dynamic behavior** of the system. They can be formalized using a time-dependent notation, like

\[ z(t) = \frac{1}{2}gt^2 \]

for the free-falling elevator
Example

In first order logic, with

- \( E(t) \) = elevator position at time \( t \)
- \( P(n,t) \) = pending request at floor \( n \) at time \( t \)
- \( S(n,t) \) = servicing of floor \( n \) at time \( t \)

Any request must ultimately be satisfied

\[
\forall t \ \forall n \ ( P(n,t) \Rightarrow \exists t' > t : S(n,t') )
\]

The elevator never traverse a floor for which a request is pending without satisfying it

\[
\forall t \forall t' > t \forall n \ (P(n,t) \land E(t') \neq n \land \exists t < t'' < t' : E(t'') = n) \Rightarrow \exists t < t'' < t' : S(n,t'')
\]
Temporal Logic

- First order logic is too cumbersome for these specifications

- Temporal logic is a logic tailored for describing properties involving time
  - the time parameter t disappears
  - temporal operators mimic linguistic constructs
    - always, until, eventually
  - the truth of a proposition depend on the state on which the system is
LTL: the language

- **Atomic propositions** \( p_1, p_2, \ldots, q, \ldots \)
  - to make statements about states of the system
  - elementary descriptions which in a given state of the system have a well-defined truth value:
    - the printer is busy
    - nice weather
    - open
    - \( x + 2 = y \)

- Their choice depend on the system considered
LTL: the language

- Boolean combinators
  - true \( \top \)
  - false \( \bot \)
  - negation \( \neg \)
  - conjunction \( \land \)
  - disjunction \( \lor \)
  - implication \( \Rightarrow \)

Note: read \( p \Rightarrow q \) as “if \( p \) then \( q \)” rather than “\( p \) implies \( q \)”.

Try \( (1 = 2) \Rightarrow \text{Sint_Klas_exists} \)
LTL: the language

- **Temporal combinators** allows to speak about the sequencing of states along a computation (rather than about states individually)

- **neXt**
  
  - $X\phi$ = *in the next state $\phi$ holds*
  
  - Examples: $XX$error and $XXX$ok

```
0 1 2 2
<table>
<thead>
<tr>
<th>warm, ok</th>
<th>ok</th>
<th>error</th>
<th>error</th>
</tr>
</thead>
</table>

0 1 2 0
| warm, ok | ok | error | warm, ok |
```


LTL: Temporal combinators

- **Future**
  - $F\phi = \text{in some future state } \phi \text{ holds}$ (at least once and without saying in which state)
  - For example, $\text{warm } \Rightarrow F\text{ok}$ holds if we are in a “warm” state then we will be in an “ok” state.

---

![Diagram](attachment:image.png)
LTL: Temporal combinators

- **Globally**
  - $G\phi = \text{in all future states } \phi \text{ always holds}$
  - It is the dual of $F$: $G\phi = \neg F \neg \phi$

- For example $G(warm \Rightarrow Fok)$ holds if at any time when we are in a “warm” state we will later be in an “ok” state.

- $G(warm \Rightarrow X\neg \text{warm})? G(ok \Rightarrow X\text{warm})?$

![Diagram showing states and transitions]

- States: 0 (warm, ok), 1 (ok), 2 (error)
- Transitions: 0 to 0, 0 to 1, 1 to 2, 2 to 1, 2 to 2
LTL: Temporal combinators

- **Until**
  \[ \square \phi_1 U \phi_2 = \phi_2 \] will hold in some future state, and in all intermediate states \( \phi_1 \) will hold.

- **Weak until**
  \[ \square \phi_1 W \phi_2 = \phi_1 \] holds in all future states until \( \phi_2 \) holds.
  \[ \square \] it may be the case \( \phi_2 \) will never hold.
LTL: Temporal combinators

- **Release**

  - $\phi_1 R \phi_2 = \phi_2$ holds in all future states up to (and including) a state when $\phi_1$ holds (if ever).
  
  - It is the dual of $U$: $\phi_1 R \phi_2 = \neg(\neg \phi_1 U \neg \phi_2)$
LTL - Priorities

- Unary connectives bind most tightly
  - $\neg$, $X,F,G$
- Next come $U$, $R$ and $W$
- Finally come $\land$, $\lor$ and $\Rightarrow$

$Fp \Rightarrow Gr \lor \neg q Up$

Diagram:

```
  F
  p

  G
  r

  U
  p

  \lor

  \neg

  q
```
LTL models: Transition Systems

- **Transition system**: \(<S, \to, L>\)
  - \(S\): set of states
  - \(L: S \to \mathcal{P}(\text{Atoms})\): labelling function
  - \(\to \subseteq S \times S\): transition relation
  - Every state \(s\) has some successor state \(s'\) with \(s \to s'\)

- A system evolves from one state to another under the action of a transition

- We label a state with propositions that hold in that state
Computation paths

- **Path**: an infinite sequence $\pi$ of states such that each consecutive pair is connected by a transition

  \[ 0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow \ldots \]

- For $i \geq 1$, we write $\pi^i$ for the suffix of a path $\pi$ starting at $i$. 
Semantics (I)

Let $M = <S, \rightarrow, L>$ be a transition system, and $\pi = s_1 \rightarrow s_2 \rightarrow \ldots$ a path of $M$.

- $\pi \models \top$  
  always
- $\pi \models p$  
  iff $p \in l(s_1)$
- $\pi \models \neg \phi$  
  iff $\pi \not\models \phi$
- $\pi \models \phi_1 \land \phi_2$  
  iff $\pi \models \phi_1$ and $\pi \models \phi_2$
Semantics (II)

- $\pi \models X\phi$ iff $\pi^2 \models \phi$
- $\pi \models F\phi$ iff there is $1 \leq i$ such that $\pi^i \models \phi$
- $\pi \models G\phi$ iff for all $1 \leq i$, $\pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ iff there is $1 \leq i$ such that $\pi^i \models \phi_2$
  and for all $j < i$, $\pi^j \models \phi_1$
- $\pi \models \phi_1 W \phi_2$ iff either $\pi \models \phi_1 U \phi_2$ or for all $1 \leq i$, $\pi^i \models \phi_2$
- $\pi \models \phi_1 R \phi_2$ iff either there is $1 \leq i$ such that $\pi^i \models \phi_1$
  and for all $j \leq i$, $\pi^j \models \phi_2$
  or for all $1 \leq k$, $\pi^k \models \phi_2$
System properties

- \( M, s \models \phi \text{ iff } \pi \models \phi \) for every path \( \pi \) of \( M \) starting from the state \( s \)

- \( M, 0 \models \text{okWerror} \)

- \( M, 0 \not\models \text{okUerror} \) (Why?)