SS 2017
Software Verification
From multimodal to temporal logic, LTL and CTL

Prof. Dr. Holger Schlingloff $^{1,2}$
Dr. Esteban Pavese $^1$

(1) Institut für Informatik der Humboldt Universität
(2) Fraunhofer Institut für offene Kommunikationssysteme FOKUS
Some Motivation

- Software correctness by construction

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Using methods and tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT-BASED specifications</td>
<td>EXECUTABLE specifications</td>
<td>DATA modelling</td>
</tr>
<tr>
<td>AMBIGUOUS drawings</td>
<td>AUTOMATIC documentation, testing, coding</td>
<td>STATE MACHINES</td>
</tr>
<tr>
<td>MANUAL coding</td>
<td>and ultimately</td>
<td>MODEL checking</td>
</tr>
<tr>
<td>EXPENSIVE testing</td>
<td>BUG-FREE software</td>
<td>QUALIFIED code generators</td>
</tr>
<tr>
<td>CONTINUOUS patching</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ESA UNCLASSIFIED - For Official Use
Recap

- Deficits of (multi-)modal logic?
- Definitions of Next-, Future-, and Global-operator?
- Which of the following are valid?
  - \( \vdash G^* \phi \rightarrow X\phi \)
  - \( \vdash F^+ \phi \rightarrow F^* \phi \)
  - \( \vdash G^* \phi \rightarrow (\phi \land G^* X\phi) \)
  - \( \vdash F^* G^* \phi \rightarrow G^* F^* \phi \)
- What is a natural model for temporal logic?
- Give a formula describing the following:
  - if \( p \), then \( q \) at the next occurrence of \( p \)
  - before the next occurrence of \( p \) there is a \( q \)
  - between any two occurrences of \( p \) there is a \( q \)
LTL (Linear Temporal Logic)

\[
\text{LTL} \ ::= \ \varnothing \ | \ \bot \ | \ (\text{LTL} \rightarrow \text{LTL}) \ | \ (\text{LTL} \cup^* \text{LTL})
\]

- Usually interpreted on natural or on linearly ordered models
- Expressively equivalent to FOL (\(<\)) on linear orders
- Arguably more intuitive to use

Let \( \mathcal{M} \) be an FSM, \( w_0 \) a state in \( \mathcal{M} \), and \( \varphi \) an LTL formula. What is the meaning of \((\mathcal{M}, w_0) \models \varphi\)?
  - interpreted in the graph?
  - interpreted in all maximal traces of \( \mathcal{M} \)?

Usually, for LTL the second alternative is chosen.

\((\mathcal{M}, w_0) \models \varphi\) iff for all maximal traces \( T \) of \( \mathcal{M} \) from \( w_0 \), \((T, w_0) \models \varphi\).
CTL (Computation Tree Logic)

• Any FSM gives rise to
  ▪ a set of maximal traces, or
  ▪ a computation tree

\[ CTL ::= \mathcal{P} \mid \bot \mid (CTL \rightarrow CTL) \mid E(CTL \cup^+ CTL) \mid A(CTL \cup^+ CTL). \]

• \( w_0 \models E(\varphi \cup^+ \psi) \) iff there exists \( w_1 > w_0 \) such that \( w_1 \models \psi \), and for all \( w_2 \in U \), if \( w_0 < w_2 < w_1 \) then \( w_2 \models \varphi \).

• \( w_0 \models A(\varphi \cup^+ \psi) \) iff for all maximal paths \( p \) from \( w_0 \) there exists \( w_1 > w_0 \) on path \( p \) such that \( w_1 \models \psi \), and for all \( w_0 < w_2 < w_1, w_2 \models \varphi \).
\[
\begin{align*}
    \text{EX } \psi & \triangleq \text{E}(\perp U^+ \psi), \\
    \text{EX } \psi & \triangleq \neg \text{AX } \neg \psi, \\
    \text{EF}^+ \psi & \triangleq \text{E}(\top U^+ \psi), \\
    \text{EG}^+ \psi & \triangleq \neg \text{AF}^+ \neg \psi, \\
    \text{E}(\varphi U^* \psi) & \triangleq (\psi \lor \varphi \land \text{E}(\varphi U^+ \psi)), \\
    \text{EF}^* \psi & \triangleq (\psi \lor \text{EF}^+ \psi), \\
    \text{EG}^* \psi & \triangleq (\psi \land \text{EG}^+ \psi),
\end{align*}
\]

\[
\begin{align*}
    \text{AX } \psi & \triangleq \text{A}(\perp U^+ \psi), \\
    \text{AX } \psi & \triangleq \neg \text{EX } \neg \psi, \\
    \text{AF}^+ \psi & \triangleq \text{A}(\top U^+ \psi), \\
    \text{AG}^+ \psi & \triangleq \neg \text{EF}^+ \neg \psi, \\
    \text{A}(\varphi U^* \psi) & \triangleq (\psi \lor \varphi \land \text{A}(\varphi U^+ \psi)), \\
    \text{AF}^* \psi & \triangleq (\psi \lor \text{AF}^+ \psi), \\
    \text{AG}^* \psi & \triangleq (\psi \land \text{AG}^+ \psi),
\end{align*}
\]
$\exists F^+(\text{started} \land \neg \text{ready})$

$\forall G^*(\text{req} \Rightarrow \exists F^+ \text{ack})$

$\forall G^+ \exists F^+ \text{stack_is_empty}$

$\forall G^+ \exists F^+ \text{restart}$

Kripke models = LTS

$F^+ G^+ p$ is not expressible in CTL

$\forall G^+ \exists F^+ p$ is not expressible in LTL

- branching time – possibility properties
- linear time – fairness properties
- CTL* subsumes both CTL and LTL
Some Temporal Formulas

- Termination: $F^* X \perp$  
- Deadlock: $EF^* AX \perp$

- Nontermination (deadlock-freedom): $G^* X T$ or $G^* F^+ T$

- Guarantee: $F^* \phi$  (where $\phi$ describes some desired states)

- Safety: $G^* \phi$  (where $\phi$ describes the admissible states)

- Invariance: $G^*(\phi \rightarrow \psi)$

- Response: $G^*(\phi \rightarrow F^* \psi)$

- Obligation: $(F^* \phi \rightarrow F^* \psi)$

- Recurrence / Livelock: $G^* F^+ \phi$ („infinitely often“)

- Persistence / Trap: $F^* G^* \phi$  (where $\phi$ describes the trap)

- Reactivity: $(G^* F^* \phi \rightarrow G^* F^* \psi)$ („infinitely many ping lead to infinitely many pong“)

- Possibility: $AG^* EF^* \phi$

- Testability: $EG^* \phi$
Example: Dining Philosophers

- Signature: $\mathcal{P} = \{\text{phil\_eating}, \text{fork\_available}, \text{phil\_hasLeftFork}, \text{phil\_hasRightFork}\}$
- Specification of desired properties

(Ph_1) “If philosopher 0 has both left and right fork, in the next moment (s)he will be eating”:

$$((\text{phil}_0 \text{hasLeftFork} \land \text{phil}_0 \text{hasRightFork}) \Rightarrow \neg\text{phil}_0 \text{eating}).$$

(Ph_2) “Whenever philosopher 0 has the left fork, in the next state (s)he will eat or drop the fork”:

$$\text{G}^* (\text{phil}_0 \text{hasLeftFork} \Rightarrow \neg(\text{phil}_0 \text{eating} \lor \text{fork}_0 \text{available})).$$

(Ph_3) “If philosopher 0 is eating, (s)he will do so until making the forks available”:

$$(\text{phil}_0 \text{eating} \Rightarrow (\text{phil}_0 \text{eating} \lor \text{fork}_0 \text{available} \land \text{fork}_1 \text{available})).$$
(Ph_4) “Philosopher 0 taking the right or philosopher 1 the left fork releases the availability of fork 0”:

(Ph_5) “Always at most one of {philosopher 0, philosopher 1} is eating”:

(Ph_6) “Philosopher 0 will eventually be eating”:

(Ph_7) “Philosopher 0 will always eventually be eating”:

(Ph_8) “Philosopher 0 can always eventually be eating”

(Ph_9) “Possibly Philosopher 0 will starve”