PSC 2020/21 (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni
http://www.di.unipi.it/~bruni/

http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/

22a - Temporal logic
Testing

how do you guarantee that your code is correct?

testing can show the presence of bugs

not their absence

coverage of all cases: difficult to achieve

especially in concurrent systems! (because of nondeterminism)
Formal logics

what does it mean to be correct? to satisfy some properties

how are these properties expressed? in some syntax

formal logics serve to express properties about programs

safety: something bad will not happen
liveness: something good will happen

model checking are certain properties satisfied (by a model of the program)?
Temporal logics

notion of time (discrete, infinite)

properties of states (atomic proposition)

*linear operators* at the next instant
always
never
eventually

*path quantifiers* (nondeterministic systems)
for all possible futures
in a possible future
Modal logics

notion of time  (discrete, infinite)

properties of states  (atomic proposition)

*modal operators*  at the next step
  at any next step
  (like HM logic)

*fix point operators*  recursively defined formulas
  minimal / maximal fixpoint
  (meaning of a formula: the set of states where it holds)
LTL
Linear temporal logic
Linear Temporal Logic

models

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots \rightarrow \bullet \rightarrow \cdots \]

syntax

\[ \psi ::= \tt \mid \ff \mid \neg \psi \mid \psi_0 \land \psi_1 \mid \psi_0 \lor \psi_1 \]

\mid p \quad \text{atomic proposition} \quad p \in P

\mid O\psi \quad \text{NEXT: } \psi \text{ holds at the next instant of time}

\mid F\psi \quad \text{FINALLY: } \psi \text{ holds sometimes in the future}

\mid G\psi \quad \text{GLOBALLY: } \psi \text{ holds always in the future}

\mid \psi_0 U \psi_1 \quad \text{UNTIL: } \psi_0 \text{ holds until } \psi_1 \text{ is true}

\text{O}\psi \text{ sometimes written } X\psi \text{ or } N\psi
Linear Structure

$S : P \rightarrow \wp(\mathbb{N})$

set of atomic propositions

$S(p)$ is the set of time instants in which $p$ holds

$S(p) = \{ n \mid p \text{ holds at } n \}$

Shift

$S^k : P \rightarrow \wp(\mathbb{N})$

$S^k(p) = \{ n - k \mid n \geq k \land n \in S(p) \}$

$S^k(p) = \{ m \mid m + k \in S(p) \}$
Example

\[ S : P \to \wp(\mathbb{N}) \]

\[ S(p) = \{0, 1, \ldots\} \quad S(q) = \{0, 2, n, \ldots\} \quad S(r) = \{n, \ldots\} \]

\[ S^2 : P \to \wp(\mathbb{N}) \]
LTL: satisfaction

$LTL \text{ formula}$

$S \models \psi$

linear structure
LTL: satisfaction

$S \models tt$

$S \models \neg \psi \quad \text{iff} \quad S \not\models \psi$

$S \models \psi_0 \land \psi_1 \quad \text{iff} \quad S \models \psi_0 \quad \text{and} \quad S \models \psi_1$

$S \models \psi_0 \lor \psi_1 \quad \text{iff} \quad S \models \psi_0 \quad \text{or} \quad S \models \psi_1$

$S \models p \quad \text{iff} \quad 0 \in S(p)$

$S \models O \psi \quad \text{iff} \quad S^1 \models \psi$

$S \models F \psi \quad \text{iff} \quad \exists k \in \mathbb{N}. \; S^k \models \psi$

$S \models G \psi \quad \text{iff} \quad \forall k \in \mathbb{N}. \; S^k \models \psi$

$S \models \psi_0 U \psi_1 \quad \text{iff} \quad \exists k \in \mathbb{N}. \; S^k \models \psi_1 \quad \text{and} \quad \forall i < k. \; S^i \models \psi_0$

current time: 0
Examples

$S \models O \ p$

$S \models F \ p$

$S \models G \ p$

$S \models p \cup q$
LTL: equivalent formulas

\( \psi_0 \equiv \psi_1 \) iff \( \forall S. S \models \psi_0 \iff S \models \psi_1 \)

\[ F \, \psi \equiv \text{tt U } \psi \]

\[ G \, \psi \equiv \neg(F \, \neg \psi) \]
\[ \equiv \neg(\text{tt U } \neg \psi) \]

\( \psi_0 \Rightarrow \psi_1 \triangleq \psi_1 \lor \neg \psi_0 \)
Examples

\[ G \neg \text{error} \]
error will never arise

\[ press \Rightarrow F \text{ error} \]
if you press now, an error will arise in the future

\[ G F \text{ enter} \]
enter will happen infinitely often (fairness)

\[ F G \text{ idle} \]
the system will stay idle from some time in the future onward

\[ G \left( req \Rightarrow (req \cup \text{eval}) \right) \]
whenever a request is made, it holds until evaluated
LTL automata-like models
LTL, again

models

\begin{tikzpicture}

\node (0) at (0,0) {$0$};
\node (1) at (1,0) {$1$};
\node (2) at (2,0) {$2$};
\node (n) at (4,0) {$n$};
\node (n+1) at (5,0) {$\ldots$};
\node (n-1) at (-1,0) {$\ldots$};

\draw[->] (0) to (1);
\draw[->] (1) to (2);
\draw[->] (2) to (n);
\draw[->] (n) to (n+1);
\draw[->] (n+1) to (n-1);
\end{tikzpicture}

syntax

$\psi ::= \text{tt} \mid \text{ff} \mid \neg \psi \mid \psi_0 \land \psi_1 \mid \psi_0 \lor \psi_1$

$p$ \hspace{1cm} \text{atomic proposition $p \in P$}

$O\psi$ \hspace{1cm} \text{NEXT: $\psi$ holds at the next instant of time}

$F\psi$ \hspace{1cm} \text{FINALLY: $\psi$ holds sometimes in the future}

$G\psi$ \hspace{1cm} \text{GLOBALLY: $\psi$ holds always in the future}

$\psi_0 U \psi_1$ \hspace{1cm} \text{UNTIL: $\psi_0$ holds until $\psi_1$ is true}

$O\psi$ sometimes written $X\psi$ or $N\psi$
Automata-like models

the formula must be satisfied along all (infinite) traces
(if we enter a deadlock state, the last state is repeated forever)
Exercise

The formula must be satisfied along all (infinite) traces

(if we enter a deadlock state, the last state is repeated forever)
Exercise

The formula must be satisfied along all (infinite) traces
(if we enter a deadlock state, the last state is repeated forever)
CTL*, CTL
Computational tree logic
Computational Tree Logic

models

syntax (CTL*)

\[\psi ::= \text{tt} | \text{ff} | \neg \psi | \psi_0 \land \psi_1 | \psi_0 \lor \psi_1 \quad \text{classical ops}\]

\[\psi ::= p | O\psi | F\psi | G\psi | \psi_0 U \psi_1 \quad \text{linear ops}\]

\[\psi ::= E\psi \quad \text{POSSIBLY: there is a path that satisfies}\]

\[\psi ::= A\psi \quad \text{ALWAYS: every path satisfies}\]
Infinite Tree

$T = (V, \rightarrow)$ directed graph

tree

$v_0 \in V$ root: a distinguished vertex (no incoming arc)

exactly one directed path from $v_0$ to any other vertex $v \in V$

infinite

every node has a child
Branching Structure

\[ T = (V, \rightarrow) \text{ infinite tree} \]

\[ S : P \rightarrow \wp(V) \]

\[ S(p) = \{ x \in V \mid x \text{ satisfies } p \} \]
Infinite Path

\[ T = (V, \rightarrow) \quad S : P \rightarrow \varnothing(V) \quad \text{branching structure} \]

infinite path \[ T = (V, \rightarrow) \quad \pi : \mathbb{N} \rightarrow V \quad (\pi = v_0 v_1 \cdots) \]

such that \( \forall k \in \mathbb{N}. \ v_k \rightarrow v_{k+1} \)

path shifting \[ \pi = v_0 v_1 \cdots \quad \pi^k = v_k v_{k+1} \cdots \]

\[ \pi : \mathbb{N} \rightarrow V \quad \pi^k : \mathbb{N} \rightarrow V \]

\[ \pi^k(i) = \pi(k + i) \]
CTL*: satisfaction

$S, \pi \models \psi$

infinite path

branching structure

CTL* formula
CTL*: satisfaction

\( S, \pi \models \mathsf{tt} \)

\( S, \pi \models \neg \psi \) \iff \( S, \pi \not\models \psi \)

\( S, \pi \models \psi_0 \land \psi_1 \) \iff \( S, \pi \models \psi_0 \) and \( S, \pi \models \psi_1 \)

\( S, \pi \models \psi_0 \lor \psi_1 \) \iff \( S, \pi \models \psi_0 \) or \( S, \pi \models \psi_1 \)

\( S, \pi \models p \) \iff \( \pi(0) \in S(p) \)

\( S, \pi \models \mathsf{O}\psi \) \iff \( S, \pi^1 \models \psi \)

\( S, \pi \models \mathsf{F}\psi \) \iff \( \exists k \in \mathbb{N}. S, \pi^k \models \psi \)

\( S, \pi \models \mathsf{G}\psi \) \iff \( \forall k \in \mathbb{N}. S, \pi^k \models \psi \)

\( S, \pi \models \psi_0 \mathsf{U}\psi_1 \) \iff \( \exists k \in \mathbb{N}. S, \pi^k \models \psi_1 \) and \( \forall i < k. S, \pi^i \models \psi_0 \)

\( S, \pi \models \mathsf{E}\psi \) \iff \( \exists \pi'. \pi'(0) = \pi(0) \) and \( S, \pi' \models \psi \)

\( S, \pi \models \mathsf{A}\psi \) \iff \( \forall \pi'. \pi'(0) = \pi(0) \) implies \( S, \pi' \models \psi \)

state ops

path ops
CTL*: equivalent formulas

\( \psi_0 \equiv \psi_1 \iff \forall S. \forall \pi. \ S, \pi \models \psi_0 \iff S, \pi \models \psi_1 \)

\[
A \ \psi \equiv \neg(E \ \neg \psi)
\]

\[
A \ A \ \psi \equiv \ A \ \psi
\]

\[
A \ E \ \psi \equiv \ E \ \psi
\]

LTL formulas as CTL* ones

\( \psi \)

\( A \ \psi \)
Examples

E O $\psi$

analogous to HML formula $\Diamond \psi$

A G $p$

$p$ holds at any reachable state

E F $p$

$p$ holds at some reachable state

A F $p$

on every path there is a state where $p$ holds

E ($p$ U $q$)

there is a path where $p$ holds until $q$
Example

A G p

\[ \text{Diagram with nodes labeled } p \text{ connected by arrows.} \]
Example

\[ EF \quad p \]

\[ \ldots \]
Example

A F $p$

...
Example

$E(p \cup q)$
CTL
CTL formulas

each path op (A/E) appears immediately before a linear op
each linear op (O/F/G/U) appears immediately after a path op

\[
\begin{align*}
E & \circ \psi \\
A & \circ \psi \\
E & F \psi \\
A & F \psi \\
E & G \psi \\
A & G \psi \\
E & (\psi_0 U \psi_1) \\
A & (\psi_0 U \psi_1) \\
A & G F \psi \\
\end{align*}
\]

CTL*, not CTL
CTL formulas

AO $\varphi$

EO $\varphi$

AG $\varphi$

EG $\varphi$

AF $\varphi$

EF $\varphi$

A[$\varphi_1$ U $\varphi_2$]

E[$\varphi_1$ U $\varphi_2$]
CTL: minimal set of ops

\[ \neg \cdot \cdot \lor \cdot \EO \cdot \EG \cdot \E(\cdot \U \cdot) \]

\[ \AO \ \psi \equiv \neg (\EO \neg \psi) \]

\[ \AF \ \psi \equiv \neg (\EG \neg \psi) \]

\[ \EF \ \psi \equiv \E (\tt \ U \ \psi) \]

\[ \AG \ \psi \equiv \neg (\EF \neg \psi) \]

\[ \equiv \neg \E (\tt \ U \neg \psi) \]

\[ \A (\psi_0 \ U \ \psi_1) \equiv \neg (\EG \neg \psi_1 \lor \E (\neg \psi_1 \ U \ \neg (\psi_0 \lor \psi_1))) \]
Expressiveness

\[(F \ G \ p) \lor (AG \ EF \ p)\]

\[\mu\]

CTL*

LTL

CTL

\[F \ G \ p\]

\[AG \ EF \ p\]