Introduction to CTL

Model verification

- The main purpose of a model checker is to verify the model with respect to a requirement specification.
- Like the model, the requirement specification must be expressed in a formally well-defined and machine readable language.
- Several such logics exist in the scientific literature
- Uppaal uses a simplified version of **CTL**.

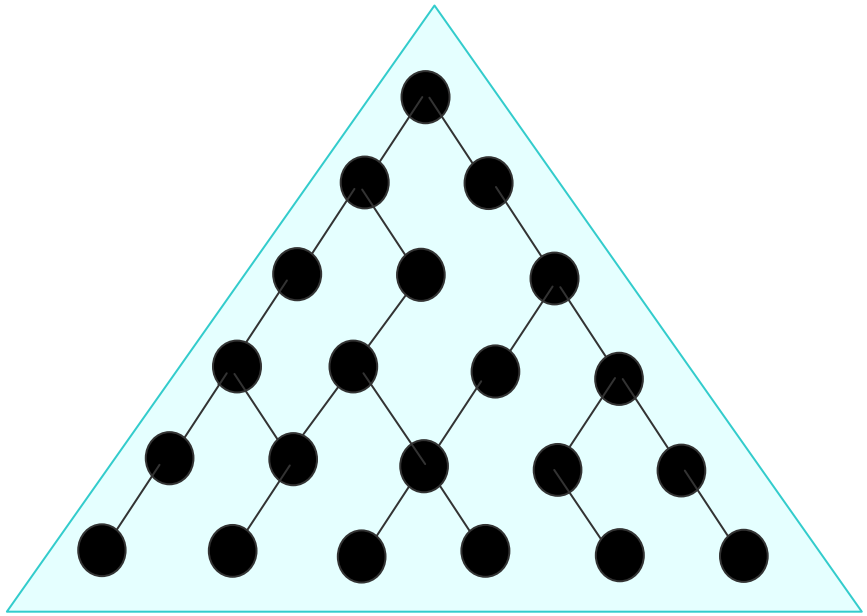
CTL

CTL: Computation Tree Logic defines about how the state of a system can evolve over time.
CTL formulas consist of the usual atomic propositional logic formulas, plus temporal connectives.
The propositional logic formulas are expressions about the state of the system. The temporal connectives are expressions about paths into the future that the state of the system can follow.
CTL consists of path formulae and state formulae.

- State formulae describe individual states
- path formulae quantify over paths or traces of the model.

CTL

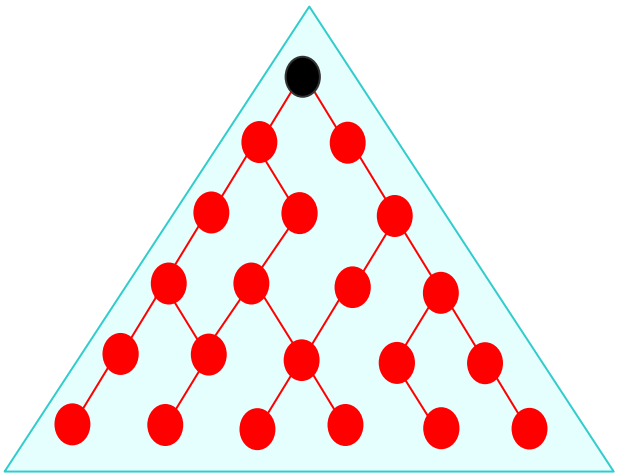
Temporal connectives are pairs of symbols. They talk about what can happen from the current state. The "current" state is the one being described in the formula. The future is infinite, i.e. the computation doesn't halt, although it can stay in the same state forever.

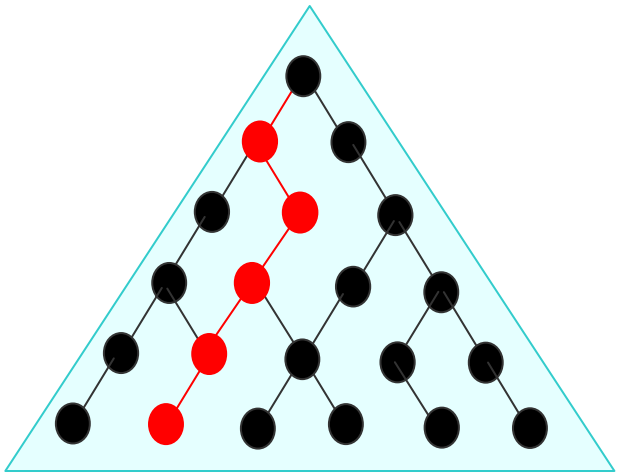


CTL

The first member of the pair is one of

A - meaning on all paths from the "current" state, read as "inevitably"
E - meaning on at least one path from the "current" state, read as "possibly"



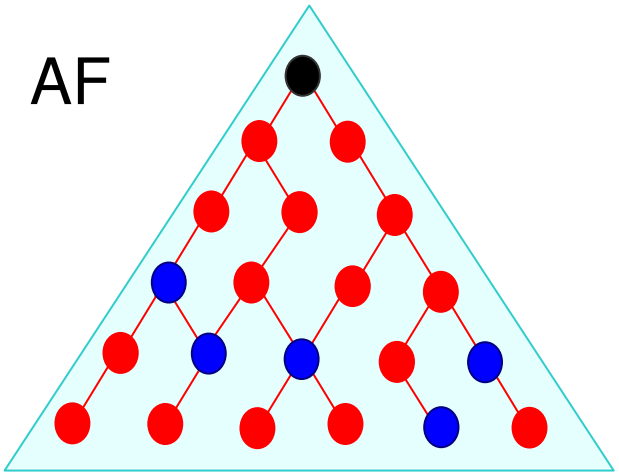


CTL

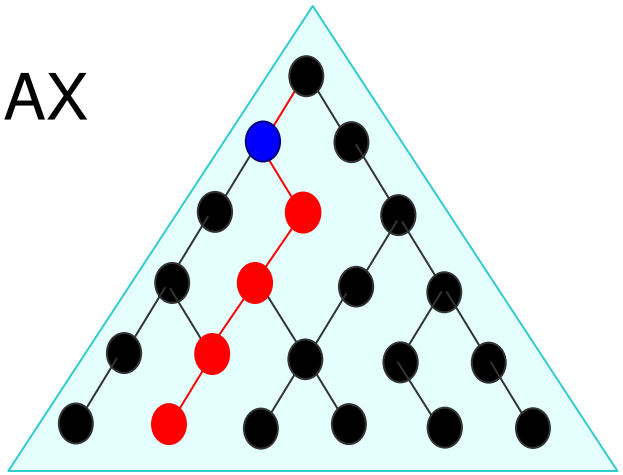
The second member of the pair is one of

X - meaning the next state
G - meaning all future states, read as "globally"
F - meaning some future state
U - meaning until

AF



AX



CTL

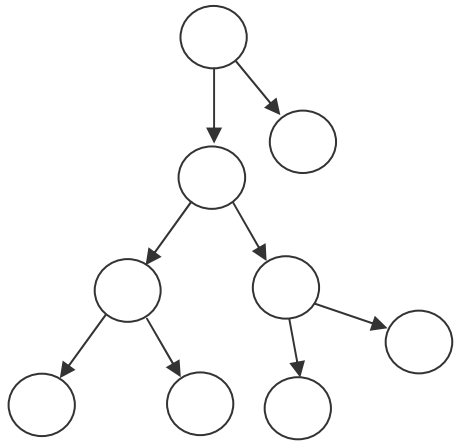
- Suppose that the system is in some state S. The future of S, by definition, includes S.
- φ is true iff it is satisfied by the current state S.
- AX (φ) is true iff φ is true for every immediate successor to state S
- AG (φ) is true iff φ is true for every successor to state S, including S That is, φ is true for all states on all paths into the future from S (*the subtree originating from S*).
- AF (φ) is true iff on all paths into the future from S, there is a state where φ holds.
- A [$\varphi U \theta$] is true iff all paths starting in state S satisfy φ until the reach a state in which θ holds.

CTL

- EX (φ) is true iff φ is true for at least one immediate successor to state S
- EG (φ) if true iff there is a path from S into the future for which φ holds for every state on
- the path, including S.
- EF (φ) is true iff there exists a path into the future from S on which there is a state where φ holds.
- E [$\varphi U \theta$] is true iff there exists a path starting in state S that satisfies φ until reaching a state in which θ holds.

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M



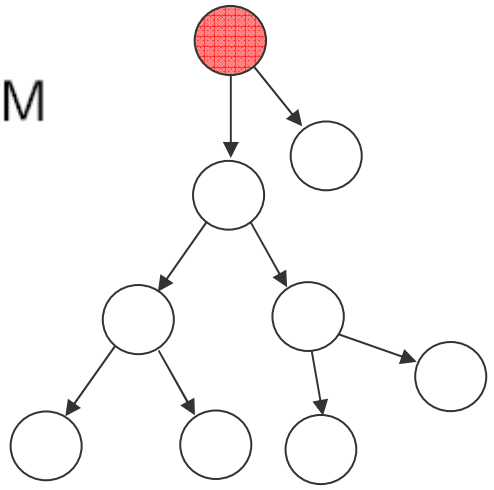
M, s_i	$\models$	a	iff	$a \in L(s_i)$	
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$	
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$	
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for all* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for all* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \psi$ *and*
					for all $i \leq k < j$: $M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \psi$ *and*
					for all $i \leq k < j$: $M, s_k \models \varphi$

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

M, s_i	$\models$	a	iff	$a \in L(s_i)$	
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$	
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$	
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s$	φ
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for all* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for all* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \psi$ *and*
					for all $i \leq k < j$: $M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i$: $M, s_j \models \psi$ *and*
					for all $i \leq k < j$: $M, s_k \models \varphi$

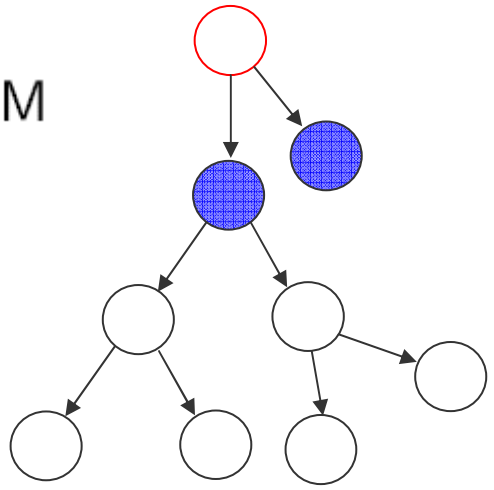
State conditions



CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

M, s_i	$\models$	a	iff	$a \in L(s_i)$	
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$	
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$	
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s_i$	$\geq i \ : M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s_i$	φ
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	$\models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \psi$ *and*
					for all $i \leq k < j \ : M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \psi$ *and*
					for all $i \leq k < j \ : M, s_k \models \varphi$



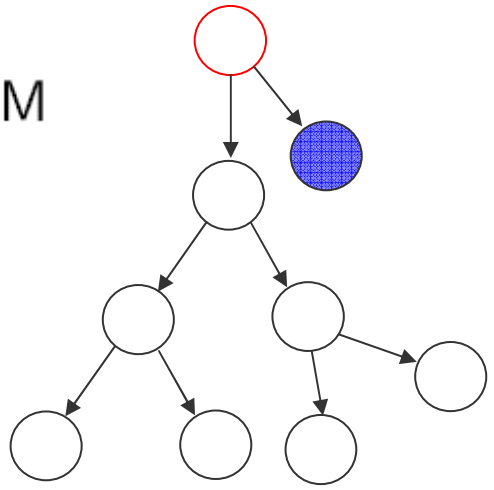
X – neXt state conditions

A – for all

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

M, s_i	$\models$	a	iff	$a \in L(s_i)$	
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$	
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$	
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	$s_{i+1} \models \varphi$
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s_i$	$\geq i \ : M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s_i$	$\geq i \ : M, s_j \models \varphi$
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	$\models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \psi$ *and*
					for all $i \leq k < j \ : M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,	*for some* $j \geq i \ : M, s_j \models \psi$ *and*
					for all $i \leq k < j \ : M, s_k \models \varphi$



X – neXt state conditions

E – exists

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

$M, s_i \models a$ iff $a \in L(s_i)$

$M, s_i \models \neg\varphi$ iff $M, s_i \not\models \varphi$

$M, s_i \models \varphi \vee \psi$ iff $M, s_i \models \varphi$ *or* $M, s_i \models \psi$

$M, s_i \models AX\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models EX\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models AG\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models EG\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models AF\varphi$ iff *for all* $(s_i, s_i$ $\geq i$: $M, s_j \models \varphi$

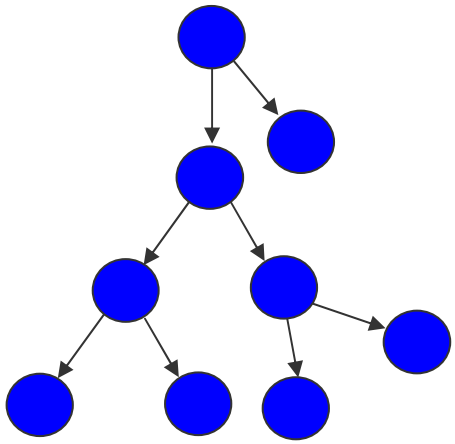
$M, s_i \models EF\varphi$ iff *for some* $(s_i$ $\geq i$: $M, s_j \models \varphi$

$M, s_i \models A(\varphi U \psi)$ iff *for all* $(s_i, s_{i+1}, \ldots)$, $\models \psi$ *and*

for all $i \leq k < j$: $M, s_k \models \varphi$

$M, s_i \models E(\varphi U \psi)$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

$\leq k < j$: $M, s_k \models \varphi$



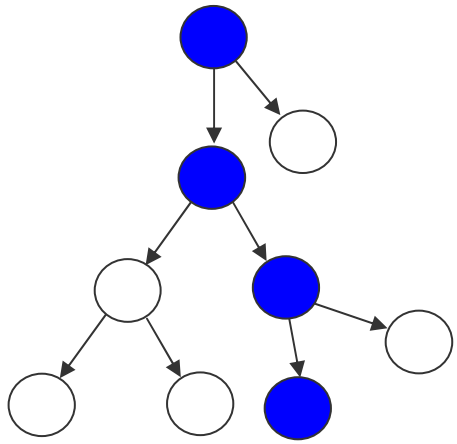
G – all states in the path

A – for all paths

G "all successors" – expressed as [] in Uppaal

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M



$M, s_i \models a$ iff $a \in L(s_i)$

$M, s_i \models \neg\varphi$ iff $M, s_i \not\models \varphi$

$M, s_i \models \varphi \vee \psi$ iff $M, s_i \models \varphi$ *or* $M, s_i \models \psi$

$M, s_i \models AX\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models EX\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models AG\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models EG\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models AF\varphi$ iff *for all* $(s_i, s_i$ … $\geq i$: $M, s_j \models \varphi$

$M, s_i \models EF\varphi$ iff *for some* $(s_i$ … $\geq i$: $M, s_j \models \varphi$

$M, s_i \models A(\varphi U \psi)$ iff *for all* $(s_i, s_{i+1}, \ldots)$, … $\models \psi$ *and*

for all $i \leq k < j$: $M, s_k \models \varphi$

$M, s_i \models E(\varphi U \psi)$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

for all $i \leq k < j$: $M, s_k \models \varphi$

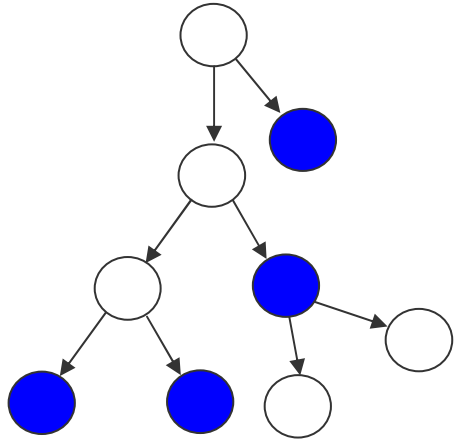
G – all states in the path

E – exists one path

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

M, s_i	$\models$	a	iff	$a \in L(s_i)$
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s$ i : $M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s$ i : $M, s_j \models \varphi$
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and* *for all* $i \leq k < j$: $M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and* $\leq k < j$: $M, s_k \models \varphi$



A – for all paths

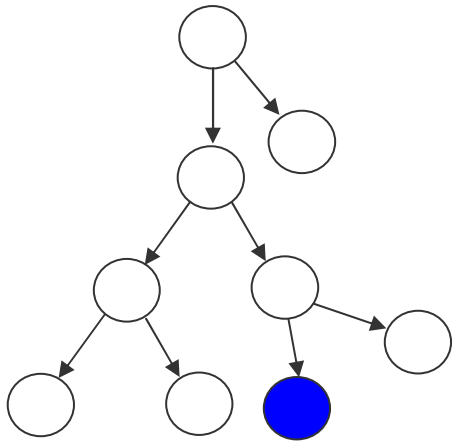
F – exists a state in the path

F "some successor"– expressed as <> in Uppaal

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

M, s_i	$\models$	a	iff	$a \in L(s_i)$
M, s_i	$\models$	$\neg\varphi$	iff	$M, s_i \not\models \varphi$
M, s_i	$\models$	$\varphi \vee \psi$	iff	$M, s_i \models \varphi$ *or* $M, s_i \models \psi$
M, s_i	$\models$	$AX\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$
M, s_i	$\models$	$EX\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$,
M, s_i	$\models$	$AG\varphi$	iff	*for all* $(s_i, s$ i : $M, s_j \models \varphi$
M, s_i	$\models$	$EG\varphi$	iff	*for some* $(s$ i : $M, s_j \models \varphi$
M, s_i	$\models$	$AF\varphi$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$EF\varphi$	iff	*for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$
M, s_i	$\models$	$A(\varphi U \psi)$	iff	*for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and* *for all* $i \leq k < j$: $M, s_k \models \varphi$
M, s_i	$\models$	$E(\varphi U \psi)$	iff	*for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and* *for all* $i \leq k < j$: $M, s_k \models \varphi$

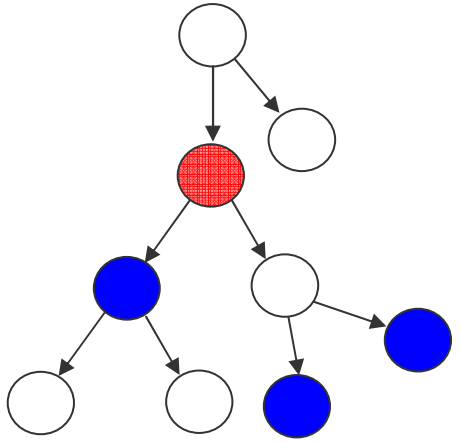


E – exists one path

F – exists a state in the path

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M



$M, s_i \models a$ iff $a \in L(s_i)$

$M, s_i \models \neg\varphi$ iff $M, s_i \not\models \varphi$

$M, s_i \models \varphi \vee \psi$ iff $M, s_i \models \varphi$ *or* $M, s_i \models \psi$

$M, s_i \models AX\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models EX\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models AG\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models EG\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models AF\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models EF\varphi$ iff *for some* $(s_i, s_{i+1},$ $M, s_j \models \varphi$

A – for all paths

$M, s_i \models A(\varphi U \psi)$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

for all $i \leq k < j$: $M, s_k \models \varphi$

$M, s_i \models E(\varphi U \psi)$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

$k < j$: $M, s_k \models \varphi$

U – “eventually” if there exists a state for which ψ, then φ for its successors

CTL

Let $(s_i, s_{i+1}, \ldots)$ be a path outgoing from state s_i in M

$M, s_i \models a$ iff $a \in L(s_i)$

$M, s_i \models \neg\varphi$ iff $M, s_i \not\models \varphi$

$M, s_i \models \varphi \vee \psi$ iff $M, s_i \models \varphi$ *or* $M, s_i \models \psi$

$M, s_i \models AX\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models EX\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, $s_{i+1} \models \varphi$

$M, s_i \models AG\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models EG\varphi$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for all* $j \geq i$: $M, s_j \models \varphi$

$M, s_i \models AF\varphi$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \varphi$

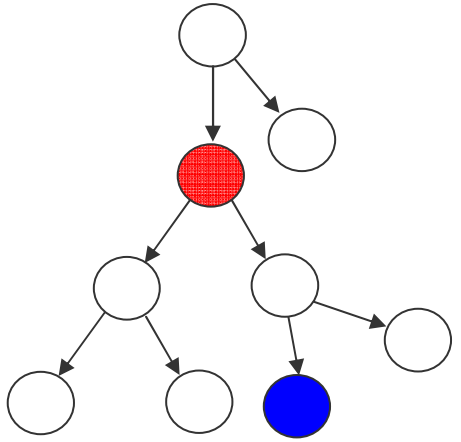
$M, s_i \models EF\varphi$ iff *for some* $(s_i, s_{i+1},$ $M, s_j \models \varphi$

$M, s_i \models A(\varphi U \psi)$ iff *for all* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

for all $i \leq k < j$: $M, s_k \models \varphi$

$M, s_i \models E(\varphi U \psi)$ iff *for some* $(s_i, s_{i+1}, \ldots)$, *for some* $j \geq i$: $M, s_j \models \psi$ *and*

$\leq k < j$: $M, s_k \models \varphi$



E – exists one path

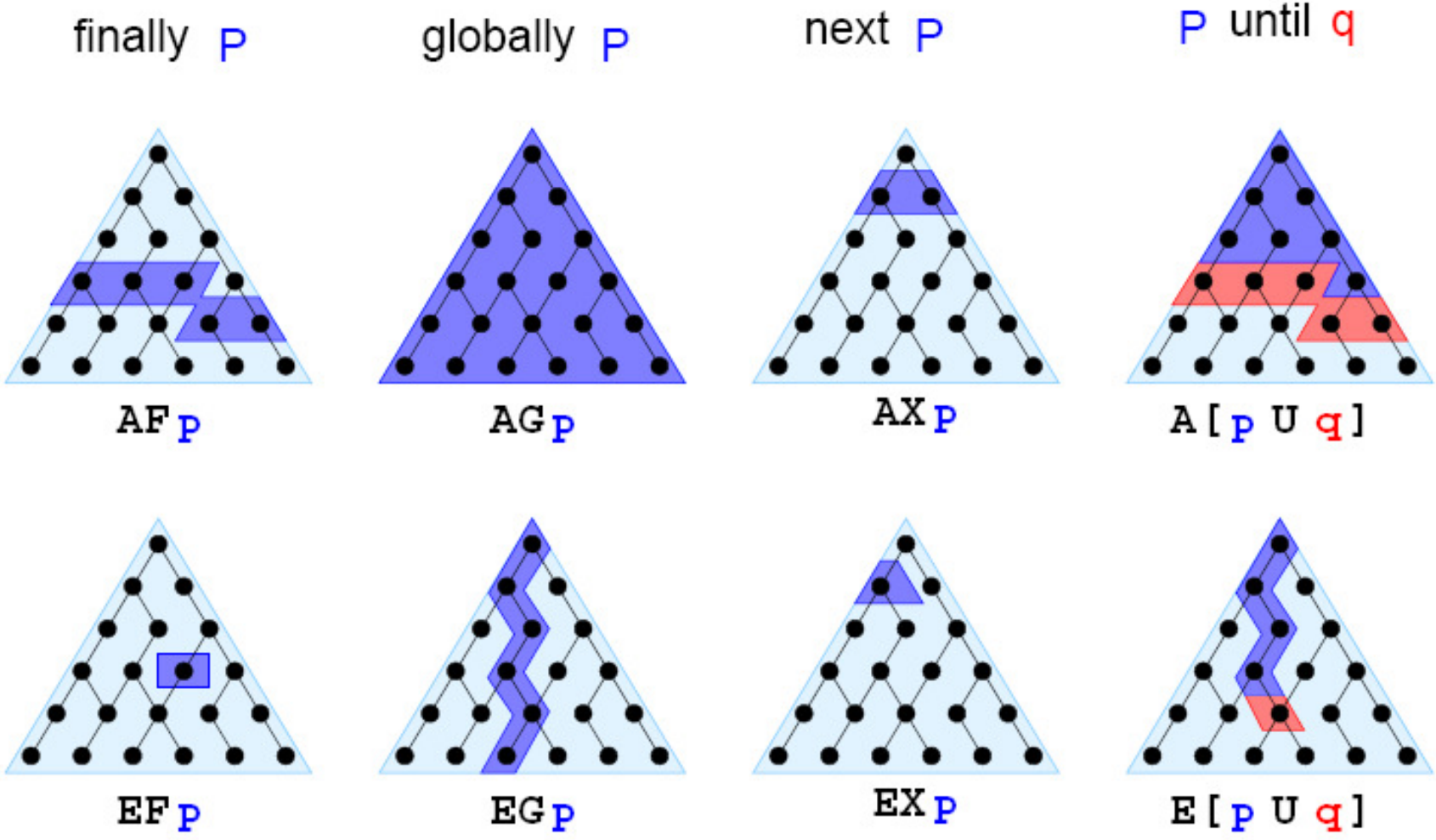
U – “eventually” if there exists a state for which ψ, then φ for its successors

CTL summary

CTL is given by the standard boolean logic enhanced with the operators $\mathbf{AX}$, $\mathbf{AG}$, $\mathbf{AF}$, $\mathbf{AU}$, $\mathbf{EX}$, $\mathbf{EG}$, $\mathbf{EF}$, $\mathbf{EU}$:

- "Necessarily Next" $\mathbf{AX}$: $\mathbf{AX}\varphi$ is true in s_t iff φ is true in every successor state s_{t+1}
- "Possibly Next" $\mathbf{EX}$: $\mathbf{EX}\varphi$ is true in s_t iff φ is true in one successor state s_{t+1}
- "Necessarily in the future" (or "Inevitably") $\mathbf{AF}$: $\mathbf{AF}\varphi$ is true in s_t iff φ is inevitably true in $\mathbf{some}$ $s_{t'}$ with $t' \geq t$
- "Possibly in the future" (or "Possibly") $\mathbf{EF}$: $\mathbf{EF}\varphi$ is true in s_t iff φ may be true in $\mathbf{some}$ $s_{t'}$ with $t' \geq t$
- "Globally" (or "always") $\mathbf{AG}$: $\mathbf{AG}\varphi$ is true in s_t iff φ is true in $\mathbf{all}$ $s_{t'}$ with $t' \geq t$
- "Possibly henceforth" $\mathbf{EG}$: $\mathbf{EG}\varphi$ is true in s_t iff φ is possibly true henceforth
- "Necessarily Until" $\mathbf{AU}$: $\mathbf{A}(\varphi \mathbf{U} \psi)$ is true in s_t iff necessarily φ holds until ψ holds.
- "Possibly Until" $\mathbf{EU}$: $\mathbf{E}(\varphi \mathbf{U} \psi)$ is true in s_t iff possibly φ holds until ψ holds.

Liveness


finally p
globally p
next p
p until q
AF p
AG p
AX p
A [p U q]
EF p
EG p
EX p
E [p U q]

CTL and Uppaal

Connectives can be combined using predicate logic

Safety: The protocol allows only one process to be in its critical section at any time.

AG ! (CS[P1] & CS[P2])

Liveness: Whenever any process want to enter its critical section, it will eventually be permitted to do so.

AG (Enter[P1] ->AF CS[P1]) & AG (Enter[P2] -> AF CS[P2])

Non-blocking: A process can always request to enter its critical section.

AG (Idle[P1] -> EX Enter[P1]) & AG (Idle[P2] -> EX Enter[P2])

CTL and Uppaal

•Like in CTL, the query language of Uppaal consists of path formulae and state formulae.

State formulae describe individual states

path formulae quantify over paths or traces of the model.

- Path formulae can be classified into
  - reachability,
  - safety and
  - liveness.

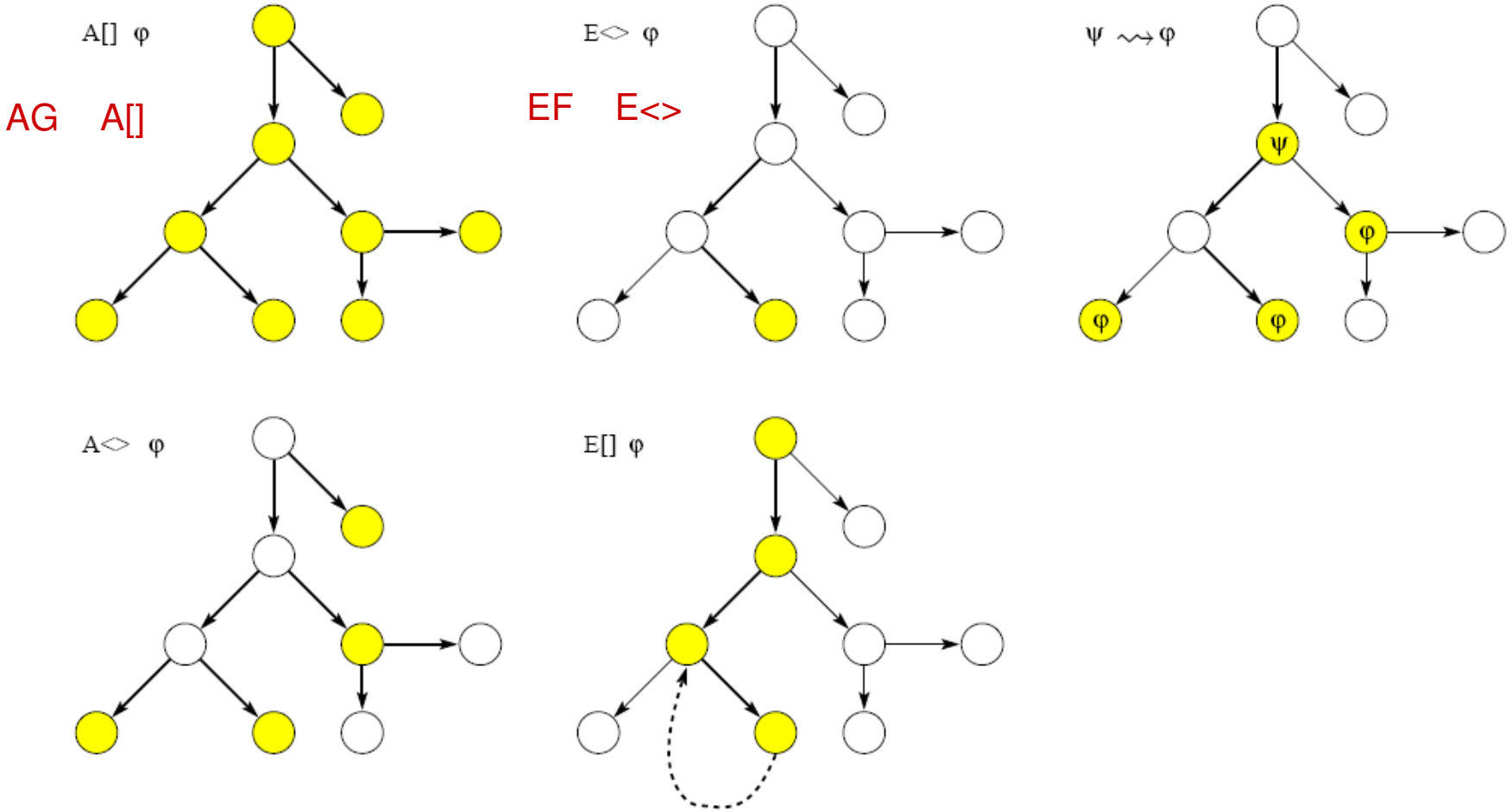
CTL and Uppaal

Derived CTL operators

potentially Φ:	$\exists\Diamond\Phi$	$=$	$\exists(\text{true}\,\mathsf{U}\,\Phi)$
inevitably Φ:	$\forall\Diamond\Phi$	$=$	$\forall(\text{true}\,\mathsf{U}\,\Phi)$
potentially always Φ:	$\exists\Box\Phi$	$:=$	$\neg\forall\Diamond\neg\Phi$
invariantly Φ:	$\forall\Box\Phi$	$=$	$\neg\exists\Diamond\neg\Phi$
weak until:	$\exists(\Phi\,\mathsf{W}\,\Psi)$	$=$	$\neg\forall\big((\Phi\wedge\neg\Psi)\,\mathsf{U}\,(\neg\Phi\wedge\neg\Psi)\big)$
	$\forall(\Phi\,\mathsf{W}\,\Psi)$	$=$	$\neg\exists\big((\Phi\wedge\neg\Psi)\,\mathsf{U}\,(\neg\Phi\wedge\neg\Psi)\big)$

CTL and Uppaal

Path formulae supported by Uppaal.



Reachability

"Something good will eventually happen"

whether a given state formula, φ, *possibly* can be satisfied by a reachable state.

Or

Does there exist a path starting at the initial state, such that φ is eventually satisfied along that path ?

We express that some state satisfying φ should be reachable using the path formula E ◊ φ.
In Uppaal, we write this property using the syntax E<> φ.

Reachability: example

For instance, when creating a model of a communication protocol involving a sender and a receiver, it makes sense to ask whether it is possible for the sender to send a message at all or whether a message can possibly be received. These properties do not by themselves guarantee the correctness of th protocol (i.e. that any message is eventually delivered), but they validate the basic behaviour of the model.

Safety properties

“something bad will never happen”.

- For instance, in a model of a nuclear power plant, a safety property might be, that the operating temperature is always (invariantly) under a certain threshold, or that a meltdown never occurs..
- For instance when playing a game, a safe state is one in which we can still win the game, hence we will possibly not loose.
- In Uppaal these properties are formulated positively, e.g., something good is invariantly true. Let φ be a state formulae. We express that φ should be true in all reachable states with the path formulae A $\Box$ φ
- … whereas E $\Box$ φ says that there should exist a maximal path such that φ is always true. In Uppaal we write A[] φ and E[] φ respectively.

Liveness

“something will eventually happen”

- when pressing the on button of the remote control of the television, then eventually the television should turn on. Or in a model of a communication protocol, any message that has been sent should eventually be received.
- In its simple form, liveness is expressed with the path formula A ◊ φ, meaning φ is eventually satisfied.
- The more useful form is the “leads to” or “response” property, written $\varphi \rightsquigarrow \psi$ which is read as whenever φ is satisfied, then eventually ψ will be satisfied, e.g. whenever a message is sent, then eventually it will be received. In Uppaal these properties are written as A<> φ and φ --> ψ, respectively.

LTL vs CTL

- LTL: Linear Temporal Logic: talks about each possible path into the future, but without considering branching.
  - I.E. we consider one path at a time and reason on it.
- LTL is CTL without the A and E connectives, except that you assume an A (all paths) connective in front of the LTL specification.

LTL vs CTL

- The LTL connectives are
- X - meaning the next state
- G - meaning all future states, read as "globally"
- F - meaning some future state
- U - meaning until

LTL vs CTL

- The temporal connectives are described below. Suppose that the system is in some state S. The future of S, by definition, includes S. Consider all the possible paths starting in S.
- phi is true iff it is satisfied by the current state S.
- X (phi) is true iff for all paths from S phi holds for the immediate successor to state S
- G (phi) is true iff for all paths from S phi holds on all states on the path
- F (phi) is true iff for all paths from S, there is a state on the path where phi holds.
- (phi U theta) is true iff for all paths from S phi holds until state occurs in which theta holds.

LTL vs CTL

- Things we may say using LTL and we cannot say in CTL:
- FG p - along every path from initial state S there is a state from which p will hold forever.
- Things we may say using CTL and we cannot say in LTL:
- AG(EF p) for all paths, in all states, there exists a path on which there is a state where p holds