Probabilistic Temporal Logics

Acknowledgement

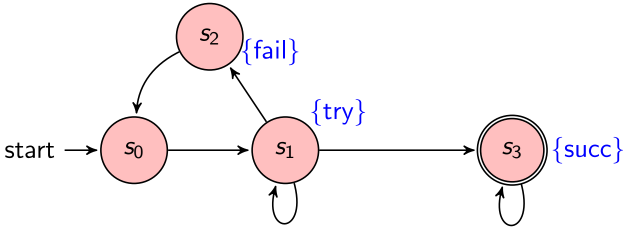
These notes are based on the fourth Dave Parker's lecture series, published on the PRISM web site: see "further reading" at the end of lecture 1.

Temporal Logics

By these we mean formal languages for reasoning about the behaviour of systems over time.

- They extend formal *propositional logic* with operators expressing temporal properties such as always(...), eventually(...)
- We need them to express formally a system model, and system properties against which the system can be tested with a software *model checker* such as SPIN, UPPAAL or PRISM
- Temporal logics have formal semantics based on *labelled (state) transition systems*.

LTS



A *labelled transition system* is a tuple $(S, s_{ini}, \rightarrow, L)$ consisting of

- S,a set of locations or *states*
- $s_{ini} \in S$, a particular *initial* state
- $\rightarrow \in S \times S$, a *transition relation*. We write $s \rightarrow s'$ rather than '$(s, s') \in \rightarrow$'.
- $L : S \rightarrow 2^{AP}$ is a function labelling each state s with a set of *atomic propositions*, possible empty.
- Drawn as a graph, with states as vertices and transitions as edges.

A DTMC, minus propabilities! Indeed, a DTMC has an underlying LTS in which $s \rightarrow s'$ iff $P(s, s') > 0$. This example is derived from a the simple communication protocol DTMC we have seen previously.

Paths in a LTS - notation

- A path is a sequence $\omega = (s_0 s_1 s_2 \ldots)$ where $\forall i \geq 0, s_i \to s_{i+1}$
  - a finite path is a finite sequence $(s_0 s_1 s_2 \ldots s_n)$ where for $0 \leq i < n, s_i \to s_{i+1}$
- $\omega(i)$ denotes state no i: s_i in the examples above.
- $\omega[\ldots i] \triangleq (s_0 s_1 s_2 \ldots s_i)$ - a (finite) *prefix* of ω
- $\omega[i \ldots] \triangleq (s_{i+1} s_{i+2} \ldots)$ - a *suffix* of ω
- (As in DTMCs) $Paths(s) \triangleq$ the set of all paths starting from s.

Computation Tree Logic - CTL

Two types of propositional formulae are defined (in terms of each other):

- *State formula:* $\varphi ::= \mathit{true} \mid \alpha \mid \varphi \wedge \varphi \mid \neg\varphi \mid \forall\psi \mid \exists\psi$
- *Path formula:* $\psi ::= \bigcirc\varphi \mid \Box\varphi \mid \Diamond\varphi \mid \varphi\, \mathcal{U}\, \varphi$
  - here, α denotes an atomic proposition, φ a state formula and ψ a path formula
- A 'CTL formula' is a state formula. A path formula is always inside scope of a $\forall$ or a $\exists$, making a state formula.

Alternative notation

- $X\varphi$ for $\bigcirc\varphi$;
- $F\varphi$ for $\Diamond\varphi$;
- $G\varphi$ for $\Box\varphi$;
- A rather than $\forall$, E rather than $\exists$

CTL Semantics in a LTS

$\varphi, \varphi_1, \varphi_2$ denote a *state* formula; ψ denotes a *path* formula ...

State formula semantics: $s \models \varphi$ means "φ is true in/of state s"

- $s \models \mathit{true}$ always
- $s \models \alpha$ iff $\alpha \in L(s)$
- $s \models \varphi_1 \wedge \varphi_2$ iff $s \models \varphi_1$ and $s \models \varphi_2$
- $s \models \neg\varphi$ iff $s \not\models \varphi$
- $s \models \forall\psi$ iff $\omega \models \psi$ for all $\omega \in \mathit{Paths}(s)$
- $s \models \exists\psi$ iff $\omega \models \psi$ for some $\omega \in \mathit{Paths}(s)$

Path formula semantics: $\omega \models \psi$ means "ψ is true of path ω"

- $\omega \models \bigcirc\varphi$ iff $\omega(1) \models \varphi$
- $\omega \models \Box\varphi$ iff for all $k \geq 0, \omega(k) \models \varphi$
- $\omega \models \Diamond\varphi$ iff for some $k \geq 0, \omega(k) \models \varphi$
- $\omega \models \varphi_1 \mathcal{U} \varphi_2$ iff for some $k \geq 0, \omega(k) \models \varphi_2$ and for all $i < k, \omega(i) \models \varphi_1$

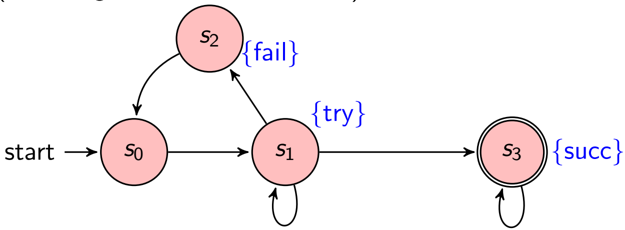
CTL Semantics (ctd)

Thus,

- $s \models \exists\Diamond\varphi$ means *some* path from s has a state on it where φ is true;
- $s \models \exists\Box\varphi$ means on *some* path from s, at every state from s on, φ is true;
- $s \models \exists(\varphi_1 \mathcal{U} \varphi_2)$ means on *some* path from s, there is a state where φ_2 is true and at previous states back to s, φ_1 is true;
- $s \models \forall\Diamond\varphi$ means *every* path from s has a state on it where φ is true;
- $s \models \forall\Box\varphi$ means on *every* path from s, at every state from s on, φ is true;
- $s \models \forall(\varphi_1 \mathcal{U} \varphi_2)$ means on *every* path from s, there is a state where φ_2 is true and at previous states back to s, φ_1 is true;

CTL Semantics Examples

(Referring to the LTS on slide 3)



Paths satisfying path formulae -

- $\omega = (s_1 s_2 \ldots)$ then $\omega \vDash \bigcirc succ$;
- $(s_0 s_1 s_1 s_3 \ldots) \vDash \neg fail \; \mathcal{U} \; succ$

States satisfying CTL (state) formulae -

- $s_1 \vDash try \wedge \neg fail$;
- $s_1 \vDash \exists \bigcirc succ$ and $s_1, s_3 \vDash \forall \bigcirc succ$;
- $s_0 \vDash \exists(\neg fail \; \mathcal{U} \; succ)$ but $s_0 \nvDash \forall(\neg fail \; \mathcal{U} \; succ)$.

Common CTL formula examples

- $\forall\Box(\neg crit_1 \wedge crit_2)$: mutual exclusion between critical sections 1, 2
- $\forall\Box\ \exists\Diamond\ init$: in every run, it is always possible to return to the initial state
- $\forall\Box(request \ \Rightarrow \forall\Diamond\ response)$: every request is always eventually responded to
  - $\Rightarrow$ denotes *implication* - see below.
- $\forall\Box\ \forall\Diamond crit_1 \wedge \forall\Box\ \forall\Diamond crit_2$: processes 1, 2 both get access to critical section infinitely often

The 'usual' form for CTL formulae has a *path quantifier* $\forall$ or $\exists$ followed by a *temporal operator* $\bigcirc$ or $\Box$ or $\Diamond$ or $(-\mathcal{U}-)$. A path formula *always* appears inside the scope of a quantifier.

Equivalences and derived operators

- *false* $\triangleq \neg$*true*
- disjuction: $\varphi_1 \vee \varphi_2 \triangleq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$
- implication: $\varphi_1 \Rightarrow \varphi_2 \triangleq \neg\varphi_1 \vee \varphi_2$ [or alternatively $\neg(\varphi_1 \wedge \neg\varphi_2)$]
- $\forall\psi \equiv \neg\exists\neg\psi$ and similarly $\exists\psi \equiv \neg\forall\neg\psi$
- $\Diamond\varphi \equiv$ *true* $\mathcal{U}\ \varphi$
- $\Box\varphi \equiv \neg\Diamond\neg\varphi$ and similarly $\Diamond\varphi \equiv \neg\Box\neg\varphi$

Some more temporal operators making path formulae -

- weak-until: $\varphi_1\mathcal{W}\varphi_2 \triangleq (\varphi_1\mathcal{U}\varphi_2) \vee \Box\varphi_1$ [alternatively $\varphi_1\mathcal{U}(\varphi_2 \vee \Box\varphi_1)$]: φ_1 holds along the path until φ_2 becomes true but this need never happen
- Thus $\varphi_1\mathcal{U}\varphi_2 \equiv (\varphi_1\mathcal{W}\varphi_2) \wedge \Diamond\varphi_2$
- *Release*: $\varphi_1\mathcal{R}\varphi_2 \triangleq \varphi_2\mathcal{W}(\varphi_1 \wedge \varphi_2)$: φ_2 holds along the path until (and including) the state where φ_1 first becomes true.

Probabilistic Computation Tree Logic - PCTL

A development of CTL: the path quantifiers $\forall, \exists$ are replaced by a 'probabilistic quantifer' $\mathbb{P}(...)$. For example,

- state formula '$\mathit{send} \Rightarrow \mathbb{P}_{\geq 0.95}(\Diamond^{\leq 10} \mathit{deliver})$'
- expresses that 'if message sent, then with probability at least 0.95, it is delivered with 10 steps'.

Again, two types of propositional formula:

- *State formula:* $\varphi ::= \mathit{true} \mid \alpha \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_{\sim p}\psi$
- *Path formula:* $\psi ::= \bigcirc\varphi \mid \varphi\, \mathcal{U}^{\leq k}\, \varphi \mid \varphi\, \mathcal{U}\, \varphi$
  - α denotes an atomic proposition, φ a state formula and ψ a path formula; $\sim$ is one of $<, >, \leq, \geq$; $p \in [0, 1]$, a probability bound; k is a positive integer.
  - Also $\mathcal{W}, \mathcal{R}$ derived from $\mathcal{U}$ as in CTL, and bounded versions $\mathcal{W}^{\leq k}, \mathcal{R}^{\leq k}$
- A 'PCTL formula' is a state formula. A path formula is always inside scope of a $\mathbb{P}$, making a state formula.

PCTL Semantics in a DTMC

$\varphi, \varphi_1, \varphi_2$ denote a *state* formula; ψ denotes a *path* formula ...

State formula semantics: $s \models \varphi$ means "φ is true in/of state s"

- $s \models \mathit{true}$ always
- $s \models \alpha$ iff $\alpha \in L(s)$
- $s \models \varphi_1 \wedge \varphi_2$ iff $s \models \varphi_1$ and $s \models \varphi_2$
- $s \models \neg\varphi$ iff $s \not\models \varphi$
- $s \models \mathbb{P}_{\sim p}\psi$ iff $\mathit{Prob}(s, \psi) \sim p$
  - where $\mathit{Prob}(s, \psi) \triangleq \mathit{Pr}_s(\{\omega \in \mathit{Paths}(s) | \omega \models \psi\})$

Path formula semantics: $\omega \models \psi$ means "ψ is true of path ω"

- $\omega \models \bigcirc\varphi$ iff $\omega(1) \models \varphi$
- $\omega \models \varphi_1 \,\mathcal{U}\, \varphi_2$ iff for some $j \geq 0, \omega(j) \models \varphi_2$ and for all $i < j, \omega(i) \models \varphi_1$
- $\omega \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ iff for some $j, 0 \leq j \leq k,\ \omega(j) \models \varphi_2$ and for all $i < j,\ \omega(i) \models \varphi_1$
- Exercise: work out the semantics for $\omega \models \varphi_1 \,\mathcal{W}^{\leq k}\, \varphi_2$ and for $\omega \models \varphi_1 \,\mathcal{R}^{\leq k}\, \varphi_2$

PCTL Semantics - commentary

- All bar $\mathbb{P}_{\sim p}$ and $\mathcal{U}^{\leq k}$ have essentially the same sematics (and same intuitive meaning) as in CTL
- The extra Markov chain structure is needed for interpreting $\mathbb{P}_{\sim p}$.
  - Recall definition of $Pr_s(\Pi)$ for a 'measurable' set Π of paths out of a state s of the DTMC.
  - Set Π is built by countible union and complementation from cylinder sets and we assign it a probability by adding up the probabilities of the cylinder sets.
  - $Pr_s(\{\omega \in Paths(s) | \omega \models \psi\})$ is the probability of the set of paths out of s satisfying path formula ψ. (This is always 'measurable').
  - Thus, for instance, $s \models \mathbb{P}_{>0.25}\ \psi$ iff the probability that ψ is true for outgoing paths from s is > 0.25.
  - $s \models \mathbb{P}_{>0.25}\ \bigcirc\ \mathit{fail}$ iff the probability is > 0.25 that *fail* will be the case pm the next state of outgoing paths from s.
- $\varphi_1\ \mathcal{U}^{\leq k}\ \varphi_2$ is a 'bounded' version of $\varphi_1\ \mathcal{U}\ \varphi_2$: at state s, it says 'φ_2 will hold within k steps (of s) and in the meantime, φ_1 holds'. In fact this is definable in CTL, not just PCTL.

Logical equivalences in PCTL

The propositional equivalences noted above for CTL obtain here also; eg

- $\neg \mathit{false} \equiv \mathit{true}$
- disjuction: $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$
- implication: $\varphi_1 \Rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$

Negation works with probabilities as you would expect; eg

- $\neg\mathbb{P}_{>p}(\varphi_1 \mathcal{U} \varphi_2) \equiv \mathbb{P}_{\leq p}(\varphi_1 \mathcal{U} \varphi_2)$
- Check the sematics of this!
- How about bounded $\mathcal{U}$?

Logical equivalences in PCTL; Reachability; Invariance

More bounded temporal operators ...

- $\Diamond^{\leq k}$, $\Box^{\leq k}$ are defined with semantics analogous to $\mathcal{U}^{\leq k}$
- ... or they can be derived: eg $\Diamond^{\leq k}\varphi \triangleq \mathit{true}\ \mathcal{U}^{\leq k}\ \varphi$
- Defining $\Box^{\leq k}\varphi \triangleq \neg\Diamond^{\leq k}\neg\varphi$ requires the ability to negate a path formula; some textbook define this, some don't.
- $\mathcal{W}^{\leq k}$ and $\mathcal{R}^{\leq k}$ derived from $\mathcal{U}^{\leq k}, \Box^{\leq k}$ similarly to slide 10

Probabilistic reachability, bounded reachability...

- $\mathbb{P}_{\sim p}\Diamond\varphi$: the probability of reaching a state satisfying φ
- $\mathbb{P}_{\sim p}\Diamond^{\leq k}\varphi$: the probability of reaching a state satisfying φ in k steps

Probabilistic (bounded) invariance ...

- $\mathbb{P}_{\sim p}\Box\varphi$: the probability of φ remaining always true
- $\mathbb{P}_{\sim p}\Box^{\leq k}\varphi$: the probability of φ remaining true for k steps

Examples

- $\mathbb{P}_{<0.05}\ \Diamond(\mathit{numFaults}/\mathit{total} > 0.1)$ - with probability less than 0.05, the fault rate is over 10%.
- $\mathbb{P}_{\geq 0.8}\ \Diamond^{\leq 15}(\mathit{numReplies} == n)$ - with probability at least 0.8, the sender has received n replies within 15 clock ticks.
- $\mathbb{P}_{<0.4}\ (\neg \mathit{fail}_A\ \mathcal{U}\ \mathit{fail}_B)$ - with probability under 0.4, component B fails before A.
- $\neg \mathit{oprtnl} \Rightarrow \mathbb{P}_{\geq 1}\ \Diamond\ \mathbb{P}_{>0.99}\ \Box^{\leq 100}\ \mathit{oprtnl}$ - if the system is not operational, then almost surely, it eventually reaches a state where it has a better than 0.99 chance of staying operational for 100 ticks.

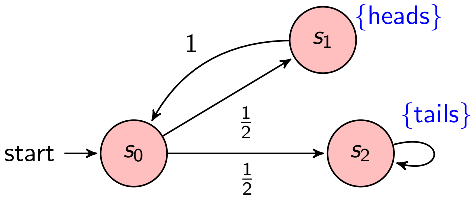
Qualitative vs quantitative properties

$\mathbb{P}$ is a kind of quantitative analogue of $\forall, \exists$ of CTL.

- $\mathbb{P}_{\sim p}\ \psi$ is *qual*itative when $p = 0$ or 1, *quant*itative when $0 < p < 1$.
- $\mathbb{P}_{>0}\ \Diamond\varphi$ is semantically equivalent to (CTL) $\exists\Diamond\varphi$
  - "There is a finite path to a state where φ holds"
  - Exercise: check this!
- $\mathbb{P}_{\geq 1}\ \Diamond\varphi$ is similar to but weaker than $\forall\Diamond\varphi$
  - "A state where φ holds is 'almost surely' reached"
  - whereas $\forall\Diamond\varphi$ says "A state where φ holds is reached"
  - For example ...

Qualitative vs quantitative properties - example

Toss a coin repeatedly until "tails"



Must "tails" always eventually be thrown?

- CTL: $\forall\Diamond tails$
- False: path $s_0 s_1 s_0 s_1 s_0 s_1 \ldots$. is a counterexample

But

- PCTL: $\mathbb{P}_{\geq 1}\Diamond tails$ is true
- ... because path $s_0 s_1 s_0 s_1 s_0 s_1 \ldots$. has probability 0.

Quantitative properties

When testing a PCTL formula $\mathbb{P}_{\sim p}\ \psi$, we might have no idea how to choose the bound, p.

- If this $\mathbb{P}$ is the outmost one in the formula, PRISM allows the form $\mathbb{P}_{=?}\ \psi$
- PRISM returns the probability of ψ
- Eg: $\mathbb{P}_{=?}\ [\Diamond(err/tot > 0.1)]$

PRISM notation

Read the chapter on Property Specification in the PRISM manual.

- PRISM supports the $\mathbb{P}$ operator in its inequality form (slide 11) as well as $\mathbb{P}_{=?}$ [...] as above.
- In fact, PRISM supports CTL as well as PCTL: remember CTL (state) formulae are true/false at a state, as is a formula of the form $\mathbb{P}_{\sim p}$ [...].
  - The CTL path quantifiers $\forall, \exists$ are denoted A, E in PRISM.
- Atomic formulae in PRISM are expressed in terms of state variables.
- A state is determined by a tuple of values of the state variables. Thus, state formulae are essentially predicates on these tuples.
- "Path properties" are path formulae in the sense of these slides. PRISM uses the notation X for $\bigcirc$; and U, W for the "until" operators we have denoted $\mathcal{U}, \mathcal{W}$. F, G are used rather than $\Diamond, \Box$.
  - $\mathbb{P}_{=?}\ [F\varphi]$ for $\mathbb{P}_{=?}\ [\Diamond\varphi]$
  - $\mathbb{P} > 0.5\ [XG^{\leq 4}\varphi]$ for $\mathbb{P}_{>0.5}\ [\bigcirc\Box^{\leq 4}\varphi]$

Rewards

PRISM provides an enhancement of basic probabilistic automaton modelling:

- A *rewards* section may be declared, in which a numerical "reward" can be assigned to each state
  - guarded by a predicate on state variables
  - may be a function of the state variables
- ... and/or to each transition, similarly.

A total reward accumulates as the model runs along a path. PRISM offers an operator $\mathbb{R}$ to query the *expected value* of reward ...

- $R \sim p[\theta]$ where $\sim$ is one of $<, \leq, >, \geq$, p is a bound, θ a "reward property"
  - true in a state of a model if "the expected reward on associated with θ of the model when starting from that state" is p
- $R = ?[\theta]$, $Rmin = ?[\theta]$, $Rmax = ?[\theta]$
  - Reports the actual expected reward value.
- Details in the Property Specification chapter of the manual

Other temporal logics?

Linear Time Logic (LTL, used by SPIN) is in some ways more expressive than CTL and its probabilistic version PCTL.

- LTL has only path formulae -
- $\psi ::= \mathit{true} \mid \alpha \mid \psi \wedge \psi \mid \neg\psi \mid \bigcirc \psi \mid \psi \, \mathcal{U} \, \psi$
  - α denotes an atomic proposition, ψ a path formula.
- Semantics: for any path ω,
  - $\omega \vDash \mathit{true}$ always
  - $\omega \vDash \alpha$ iff $\alpha \in L(\omega(0))$
  - $\omega \vDash \psi_1 \wedge \psi_2$ iff $\omega \vDash \psi_1$ and $\omega \vDash \psi_2$
  - $\omega \vDash \neg\psi$ iff $\omega \nvDash \psi$
  - $\omega \vDash \bigcirc\psi$ iff $\omega[1...] \vDash \psi$
  - $\omega \vDash \psi_1 \mathcal{U} \psi_2$ iff
    for some $n > 0$, $\omega[n...] \vDash \psi_2$ and for all $0 \leq k < n$, $\omega[k...] \vDash \psi_1$
- Derived operators as in CTL: eg propositional $\vee, \Rightarrow$ and
  - $\Diamond\psi \triangleq \mathit{true} \, \mathcal{U} \, \psi$
  - $\Box\psi \triangleq \neg\Diamond\neg\psi$
- An LTS satisfies a (path) formula iff all paths from its initial state satisfy it.

Other temporal logics?

LTL has a simpler time model (linear rather than branching) than (P)CTL but in some ways is more expressive.

- The LTL formula $\Diamond(\mathit{reqst} \wedge \bigcirc \mathit{ack})$: "Eventually there is a request followed immediately by an acknowledgement"
- ... cannot be expressed in CTL
- PCTL is essentially limited to properties than can be put in the form "B can reached via states in C [within k steps] (where B, C are sets of states).
- In (P)CTL, every temporal operator has to be within immendiate scope of a quantifer, but in LTL, temporal operators can be combined: $\Box\Diamond\psi$ and so forth.

Other temporal logics?

One idea is to add probabilities to LTL: taking cue from PCTL:

- $Prob(s, \psi) \triangleq Pr_s(\{\omega \in Paths(s) | \omega \models \psi\})$

One can then express in 'LTL+probability',

- Repeated reachability: $Prob(s, \Box\Diamond action)$
  - the probability that the action occurs infinitely often
- Persistence: $Prob(s, \Diamond\Box steadyState)$
  - the probability that the algorithm eventually reaches a steady state

PCTL*

An extension of PCTL with elements of LTL

- Maintains distict state and path formulae;
- *State formula:* $\varphi ::= \mathit{true} \mid \alpha \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_{\sim p}\psi$
- *Path formula:* $\psi ::= \varphi \mid \psi \wedge \psi \mid \neg\psi \mid \psi \, \mathcal{U}^{\leq k} \, \psi \mid \psi \, \mathcal{U} \, \psi$
  - α denotes an atomic proposition, φ a state formula and ψ a path formula;
- allows conjuction and negation (hence all boolean combinations) of path formulae; and a state formula *is* a path formula
- A PCTL* formula is a state formula (so path formulae have to be quantified)
- Example: $\mathbb{P}_{>0.1}[\Box\Diamond \mathit{critSect}_1] \wedge \mathbb{P}_{>0.1}[\Box\Diamond \mathit{critSect}_2]$