Linear-time Temporal Logic

We are interested in temporal properties of program behaviour, i.e. properties over state sequences, like that some "bad" state (expressed as a state property) is never reached, or that some state property holds until some other state property becomes true, etc.

Our theoretical set-up will be:

Models:	transition systems $\mathcal{M}$
Specs:	LTL formulas ϕ
Correctness:	$\mathcal{M}, s \models \phi$
Verification:	automata-based, language inclusion

On the practical side we will use the Spin model checker.

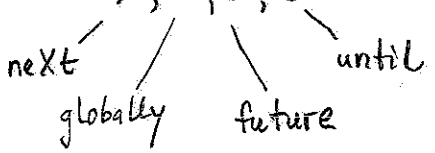
LTL Syntax

The logic is built on top of a set of atomic propositions Atoms, used to denote state properties. The logic uses

propositional constants: $\top, \bot$ atoms: $p \in Atoms$

propositional connectives: $\neg, \wedge, \vee, \rightarrow$

temporal connectives: X, G, F, U



Example formulas:

$$Fp,\ GFp,\ G(p \rightarrow Fq),\ pUq$$

The formulas of LTL are defined inductively as follows:

$$\phi ::= \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \to \phi) \mid$$
$$(X\phi) \mid (F\phi) \mid (G\phi) \mid (\phi U \phi)$$
$$\bigcirc \qquad \Diamond \qquad \Box$$

Binding convention:

strongest: $\neg, X, F, G$
next: U
then: $\wedge, \vee$
weakest: $\to$

So, $GFp \to F(q \vee s)$ abbreviates $((G(Fp)) \to (F(q \vee s)))$.

The notions of parse tree and subformula are standard.

LTL Semantics

Models for LTL-formulas are structures called transition systems, or Kripke structures, or just models.

Definition A transition system is a tuple $\mathcal{M} = (S, \to, L)$

(i) S is a set of states,
(ii) $\to \subseteq S \times S$ is a transition relation,
(iii) $L: S \to 2^{Atoms}$ is a labelling function.

In the context of LTL we usually require every state to have at least one successor state, since we are only considering infinite state sequences.

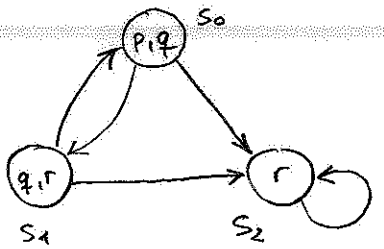
Models with a small number of states are easily presented graphically; for example:

Atoms $= \{p, q, r\}$

$S = \{s_0, s_1, s_2\}$ $\qquad \mathcal{M}$

$\to = \{(s_0, s_1), (s_0, s_2), (s_1, s_0), (s_1, s_2), (s_2, s_2)\}$

$L = \{(s_0, \{p, q\}), (s_1, \{q, r\}), (s_2, \{r\})\}$



A path of $\mathcal{M} = (S, \to, L)$ is an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ such that $s_i \to s_{i+1}$ for all $i \geq 0$.
We denote by $\pi(i)$ the i-th element s_i of π,
and by π^i the i-th suffix $s_i s_{i+1} s_{i+2} \ldots$ of π.

How do the paths /runs/ executions of the above example model look like, starting from s_0? SCCs.

The formal semantics of LTL is given in terms of a satisfaction relation $\mathcal{M}, \pi \models \phi$ inducing another satisfaction relation $\mathcal{M}, s \models \phi$.

Definition Let $\mathcal{M} = (S, \to, L)$ be a model, and let $\pi = s_0 s_1 s_2 \ldots$ be a path of $\mathcal{M}$. The satisfaction relation $\pi \models \phi$ is defined inductively as follows: (incomplete)

$$\pi \models p \overset{\text{def}}{\Leftrightarrow} p \in L(\pi(0))$$

$$\pi \models X\phi \overset{\text{def}}{\Leftrightarrow} \pi^1 \models \phi$$

$$\pi \models G\phi \overset{\text{def}}{\Leftrightarrow} \forall i \geq 0.\ \pi^i \models \phi$$

$$\pi \models F\phi \overset{\text{def}}{\Leftrightarrow} \exists i \geq 0.\ \pi^i \models \phi$$

$$\pi \models \phi U \psi \overset{\text{def}}{\Leftrightarrow} \exists i \geq 0.\ (\pi^i \models \psi \text{ and } \forall j < i.\ \pi^j \models \phi)$$

We say that model $\mathcal{M}$ satisfies formula ϕ at state $s \in S$, denoted $\mathcal{M}, s \models \phi$, if all paths π of $\mathcal{M}$ starting at s satisfy ϕ, i.e. $\mathcal{M}, \pi \models \phi$ whenever $\pi(0) = s$.

As an example, consider the model above, and the formulas

$$q,\ r,\ Xq,\ Xr,\ Gq,\ G(q \vee r),\ Fq,\ Fr,$$
$$G(r \rightarrow Gr),\ Gq \vee Gr,\ GF(p \vee r),\ pUr,\ qUr$$

all at s_0. Find counter-examples in the form of paths.

An important observation about the semantics is that the significant information of a path is the sequence of state valuations rather than the states themselves. So, we can define the notions valuation $v \subseteq Atoms$ and valuation sequence $\rho = v_0 v_1 v_2 \ldots$ and define satisfaction $\mathcal{M}, \rho \models \phi$ as:

$$\rho \models p \overset{def}{\Leftrightarrow} p \in \rho(0) \qquad \text{(only difference)}$$

Then, satisfaction $\mathcal{M}, \pi \models \phi$ can be viewed as a derived notion:

$$\mathcal{M}, \pi \models \phi \overset{def}{\Leftrightarrow} \mathcal{M}, \rho_\pi \models \phi$$

where ρ_π is the valuation sequence induced by π via L.

Here are some practical specification patterns:

G (request $\rightarrow$ F response)	every request is eventually served
GF enabled	infinitely often enabled
FG deadlock	eventually stable deadlock
GF enabled $\rightarrow$ GF running	
G (start $\rightarrow$ X (running U stop))	

LTL equivalences

We say that two LTL formulas ϕ and ψ are semantically equivalent, denoted $\phi \equiv \psi$, if

$$\mathcal{M}, \pi \models \phi \Leftrightarrow \mathcal{M}, \pi \models \psi$$

in every path π of every model $\mathcal{M}$.

$$\neg G\phi \equiv F\neg\phi \qquad \neg F\phi \equiv G\neg\phi \qquad \neg X\phi \equiv X\neg\phi$$

$$G(\phi \wedge \psi) \equiv G\phi \wedge G\psi \qquad F(\phi \vee \psi) \equiv F\phi \vee F\psi$$

$$G\phi \equiv \neg(T\,U\neg\phi) \qquad F\phi \equiv T\,U\,\phi$$

Here is a proof of $G\phi \equiv \neg(T\,U\neg\phi)$ by referring to the formal semantics of LTL: Let π be an arbitrary path of an arbitrary model $\mathcal{M}$. Then

$$\begin{aligned}
\pi \models \neg(T\,U\neg\phi) &\Leftrightarrow \text{not } \pi \models T\,U\neg\phi \\
&\Leftrightarrow \text{not } \exists i \geq 0.\ (\pi^i \models \neg\phi \text{ and } \forall j < i.\ \pi^j \models T) \\
&\Leftrightarrow \text{not } \exists i \geq 0.\ \pi^i \models \neg\phi \\
&\Leftrightarrow \text{not } \exists i \geq 0.\ (\text{not } \pi^i \models \phi) \\
&\Leftrightarrow \forall i \geq 0.\ \pi^i \models \phi \\
&\Leftrightarrow \pi \models G\phi
\end{aligned}$$

Minimal presentation of LTL: $\neg, \vee, X, U$

$$G\phi \equiv \phi \wedge XG\phi \qquad F\phi \equiv \phi \vee XF\phi \qquad \phi U\psi \equiv \psi \vee (\phi \wedge X(\phi U\psi))$$