Linear Temporal Logic

Safety vs. Liveness

- Safety : *something bad never happens*

  A counterexample is an finite execution leading to something bad happening (e.g. an assertion violation).

- Liveness : *something good eventually happens*

  A counterexample is an infinite execution on which nothing good happens (e.g. the program does not terminate).

Verification of Reactive Systems

- Classical verification à la Floyd-Hoare considered three problems:
  - Partial Correctness :
    $$\{\varphi\}\ \mathbf{P}\ \{\psi\} \text{ iff for any } s \models \varphi, \text{ if } P \text{ terminates on } s, \text{ then } P(s) \models \psi$$
  - Total Correctness :
    $$\{\varphi\}\ \mathbf{P}\ \{\psi\} \text{ iff for any } s \models \varphi,\ P \text{ terminates on } s \text{ and } P(s) \models \psi$$
  - Termination :
    $$P \text{ terminates on } s$$

- Need to reason about infinite computations :
  - systems that are in continuous interaction with their environment
  - servers, control systems, etc.
  - e.g. *"every request is eventually answered"*

Reasoning about infinite sequences of states

- Linear Temporal Logic is interpreted on infinite sequences of states
- Each state in the sequence gives an interpretation to the atomic propositions
- Temporal operators indicate in which states a formula should be interpreted

Example 1 *Consider the sequence of states:*

$$\{p, q\}\ \{\neg p, \neg q\}\ (\{\neg p, q\}\ \{p, q\})^{\omega}$$

Starting from position 2*,* q *holds forever.* $\square$

Kripke Structures

Let $\mathcal{P} = \{p, q, r, \ldots\}$ be a finite alphabet of *atomic propositions*.

A *Kripke structure* is a tuple $K = \langle S, s_0, \rightarrow, L \rangle$ where:

- S is a set of *states*,
- $s_0 \in S$ a designated *initial state*,
- $\rightarrow : \ S \times S$ is a *transition relation*,
- $L : S \rightarrow 2^{\mathcal{P}}$ is a *labeling function*.

Paths in Kripke Structures

A *path* in K is an infinite sequence $\pi : s_0, s_1, s_2 \ldots$ such that, for all $i \geq 0$, we have $s_i \to s_{i+1}$.

By $\pi(i)$ we denote the i-th state on the path.

By π_i we denote the suffix $s_i, s_{i+1}, s_{i+2} \ldots$.

$$\boxed{\inf(\pi) = \{s \in S \mid s \text{ appears infinitely often on } \pi\}}$$

If S is finite and π is infinite, then $\inf(\pi) \neq \emptyset$.

Linear Temporal Logic: Syntax

The alphabet of LTL is composed of:

- atomic proposition symbols $p, q, r, \ldots,$
- boolean connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow,$
- temporal connectives $\bigcirc, \Box, \Diamond, \mathcal{U}, \mathcal{R}.$

The set of LTL formulae is defined inductively, as follows:

- any atomic proposition is a formula,
- if φ and ψ are formulae, then $\neg\varphi$ and $\varphi \bullet \psi$, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ are also formulae.
- if φ and ψ are formulae, then $\bigcirc\varphi$, $\Box\varphi$, $\Diamond\varphi$, $\varphi\mathcal{U}\psi$ and $\varphi\mathcal{R}\psi$ are formulae,
- nothing else is a formula.

Temporal Operators

- $\bigcirc$ is read at the next time (in the next state)
- $\Box$ is read always in the future (in all future states)
- $\Diamond$ is read eventually (in some future state)
- $\mathcal{U}$ is read until
- $\mathcal{R}$ is read releases

Linear Temporal Logic: Semantics

$$K, \pi \models p \iff p \in L(\pi(0))$$

$$K, \pi \models \neg\varphi \iff K, \pi \not\models \varphi$$

$$K, \pi \models \varphi \wedge \psi \iff K, \pi \models \varphi \text{ and } K, \pi \models \psi$$

$$K, \pi \models \bigcirc\varphi \iff K, \pi_1 \models \varphi$$

$$K, \pi \models \varphi \mathcal{U} \psi \iff \text{there exists } k \in \mathbb{N} \text{ such that } K, \pi_k \models \psi \text{ and } K, \pi_i \models \varphi \text{ for all } 0 \leq i < k$$

Derived meanings:

$$K, \pi \models \Diamond\varphi \iff K, \pi \models \top \mathcal{U} \varphi$$

$$K, \pi \models \Box\varphi \iff K, \pi \models \neg\Diamond\neg\varphi$$

$$K, \pi \models \varphi \mathcal{R} \psi \iff K, \pi \models \neg(\neg\varphi \mathcal{U} \neg\psi)$$

Examples

- p holds throughout the execution of the system (p is invariant) : $\Box p$
- whenever p holds, q is bound to hold in the future : $\Box(p \rightarrow \Diamond q)$
- p holds infinitely often : $\Box\Diamond p$
- p holds forever starting from a certain point in the future : $\Diamond\Box p$
- $\Box(p \rightarrow \bigcirc(\neg q \mathcal{U} r))$ holds in all sequences such that if p is true in a state, then q remains false from the next state and until the first state where r is true, which must occur.
- $p\mathcal{R}q$: q is true unless this obligation is released by p being true in a previous state.

LTL $\equiv$ FOL

Theorem 1 *LTL and FOL on infinite words have the same expressive power.*

From LTL to FOL:

$$
\begin{aligned}
Tr(q) &= p_q(t) \\
Tr(\neg\varphi) &= \neg Tr(\varphi) \\
Tr(\varphi \wedge \psi) &= Tr(\varphi) \wedge Tr(\psi) \\
Tr(\bigcirc\varphi) &= Tr(\varphi)[t+1/t] \\
Tr(\varphi \mathcal{U} \psi) &= \exists x \,.\, Tr(\psi)[x/t] \wedge \forall y \,.\, y < x \rightarrow Tr(\varphi)[y/t]
\end{aligned}
$$

The direction from FOL to LTL is done using *star-free* sets.

LTL $<$ S1S

Definition 1 *A language $L \subseteq \Sigma^\omega$ is said to be* non-counting *iff:*

$$\exists n_0 \forall n \geq n_0 \forall u, v \in \Sigma^* \forall \beta \in \Sigma^\omega \ . \ uv^n\beta \in L \iff uv^{n+1}\beta \in L$$

Example 2 0^*1^ω *is non-counting. Let $n_0 = 2$. We have three cases:*

1. $u, v \in 0^*$ *and* $\beta \in 0^*1^\omega$:

$$\forall n \geq n_0 \ . \ uv^n\beta \in L$$

2. $u \in 0^*$, $v \in 0^*1^*$ *and* $\beta \in 1^\omega$:

$$\forall n \geq n_0 \ . \ uv^n\beta \notin L$$

3. $u \in 0^*1^*$, $v \in 1^*$ *and* $\beta \in 1^\omega$:

$$\forall n \geq n_0 \ . \ uv^n\beta \in L$$

$\square$

LTL < S1S

Conversely, a language $L \subseteq \Sigma^\omega$ is said to be *counting* iff:

$$\forall n_0 \exists n \geq n_0 \exists u, v \in \Sigma^* \exists \beta \in \Sigma^\omega \,.\, (uv^n\beta \notin L \wedge uv^{n+1}\beta \in L) \vee (uv^n\beta \in L \wedge uv^{n+1}\beta \notin L)$$

Example 3 $(00)^*1^\omega$ *is counting.*

Given n_0 *take the next even number* $n \geq n_0$, $u = \epsilon$, $v = 0$ *and* $\beta = 1^\omega$.
Then $uv^n\beta \in (00)^*1^\omega$ *and* $uv^{n+1}\beta \notin (00)^*1^\omega$. $\square$

LTL < S1S

Proposition 1 *Each LTL-definable ω-language is non-counting.*

$$\boxed{\exists n_0 \forall n \geq n_0 \forall u, v \in \Sigma^* \forall \beta \in \Sigma^\omega \; . \; uv^n\beta \models \varphi \iff uv^{n+1}\beta \models \varphi}$$

By induction on the structure of φ :

- $\varphi = a$: choose $n_0 = 1$.
- $\varphi = \neg\psi$: choose the same n_0 as for ψ.
- $\varphi = \psi_1 \wedge \psi_2$: let n_1 for ψ_1 and n_2 for ψ_2, and choose $n_0 = \max(n_1, n_2)$.

LTL < S1S

- $\varphi = \bigcirc\psi$: let n_1 for ψ and choose $n_0 = n_1 + 1$.
  - we show $\forall n \geq n_0 \ . \ (uv^n\beta)_1 \models \psi \equiv (uv^{n+1}\beta)_1 \models \psi$
  - case $u \neq \epsilon$, i.e. $u = au'$:

$$
\begin{aligned}
(au'v^n\beta)_1 \models \psi \iff u'v^n\beta \models \psi \iff \\
u'v^{n+1}\beta \models \psi \iff (au'v^{n+1}\beta)_1 \models \psi
\end{aligned}
$$

  - case $u = \epsilon$, $v = av'$:

$$
\begin{aligned}
((av')^n\beta)_1 \models \psi \iff v'(av')^{n-1}\beta \models \psi \iff \\
v'(av')^n\beta \models \psi \iff ((av')^{n+1}\beta)_1 \models \psi
\end{aligned}
$$

LTL $<$ S1S

- $\varphi = \psi_1 \mathcal{U} \psi_2$: let n_1 for ψ_1 and n_2 for ψ_2, and choose $n_0 = \max(n_1, n_2) + 1$.
  - we show $\forall n \geq n_0 \;.\; uv^n\beta \models \psi_1 \mathcal{U} \psi_2 \Rightarrow uv^{n+1}\beta \models \psi_1 \mathcal{U} \psi_2$
  - we have $(uv^n\beta)_j \models \psi_2$ and $\forall i < j \;.\; (uv^n\beta)_i \models \psi_1$ for some $j \geq 0$
  - case $j \leq |u|$: $(uv^{n+1}\beta)_j \models \psi_2$ and $\forall i < j \;.\; (uv^{n+1}\beta)_i \models \psi_1$
  - case $j > |u|$: let $j' = j + |v|$
    * $(uv^{n+1}\beta)_{j'} = (uv^n\beta)_j \models \psi_2$
    * for all $|u| + |v| \leq i < j + |v| \;.\; (uv^{n+1}\beta)_i = (uv^n\beta)_{i-|v|} \models \psi_1$
    * for all $i < |u| + |v| \;.\; ((uv)v^n\beta)_i \models \psi_1 \Leftarrow ((uv)v^{n-1}\beta)_i \models \psi_1$
  - the direction $\Leftarrow$ is left to the reader.

Theorem 2 *LTL is strictly less expressive than S1S.*

LTL Model Checking

System verification using LTL

- Let K be a model of a reactive system (finite computations can be turned into infinite ones by repeating the last state infinitely often)

- Given an LTL formula φ over a set of atomic propositions $\mathcal{P}$, specifying all bad behaviors, we build a Büchi automaton A_φ that accepts all sequences over $2^{\mathcal{P}}$ satisfying φ.

  Q: Since LTL $\subset$ S1S, this automaton can be built, so why bother?

- Check whether $\mathcal{L}(A_\varphi) \cap \mathcal{L}(K) = \emptyset$. In case it is not, we obtain a counterexample.

Generalized Büchi Automata

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet.

A *generalized Büchi automaton* (GBA) over Σ is $A = \langle S, I, T, \mathcal{F} \rangle$, where:

- S is a finite set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- $T \subseteq S \times \Sigma \times S$ is a *transition relation*,
- $\mathcal{F} = \{F_1, \ldots, F_k\} \subseteq 2^S$ is a set of *sets of final states*.

A run π of a GBA is said to be *accepting* iff, for all $1 \leq i \leq k$, we have

$$\inf(\pi) \cap F_i \neq \emptyset$$

GBA and BA are equivalent

Let $A = \langle S, I, T, \mathcal{F} \rangle$, where $\mathcal{F} = \{F_1, \ldots, F_k\}$.

Build $A' = \langle S', I', T', F' \rangle$:

- $S' = S \times \{1, \ldots, k\}$,
- $I' = I \times \{1\}$,
- $(\langle s, i \rangle, a, \langle t, j \rangle) \in T'$ iff $(s, t) \in T$ and:
  - $j = i$ if $s \notin F_i$,
  - $j = (i \mod k) + 1$ if $s \in F_i$.
- $F' = F_1 \times \{1\}$.

The idea of the construction

Let $K = \langle S, s_0, \rightarrow, L \rangle$ be a Kripke structure over a set of atomic propositions $\mathcal{P}$, $\pi : \mathbb{N} \rightarrow S$ be an infinite path through K, and φ be an LTL formula.

To determine whether $K, \pi \models \varphi$, we label π with sets of subformulae of φ in a way that is compatible with LTL semantics.

Closure

Let φ be an LTL formula written in negation normal form.

The *closure* of φ is the set $Cl(\varphi) \in 2^{\mathcal{L}(LTL)}$:

- $\varphi \in Cl(\varphi)$
- $\bigcirc\psi \in Cl(\varphi) \Rightarrow \psi \in Cl(\varphi)$
- $\psi_1 \bullet \psi_2 \in Cl(\varphi) \Rightarrow \psi_1, \psi_2 \in Cl(\varphi)$, for all $\bullet \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$.

Example 4 $Cl(\Diamond p) = Cl(\top \mathcal{U} p) = \{\Diamond p, p, \top\}\square$

Q: What is the size of the closure relative to the size of φ ?

Labeling rules

Given $\pi : \mathbb{N} \rightarrow 2^{\mathcal{P}}$ and φ, we define $\tau : \mathbb{N} \rightarrow 2^{Cl(\varphi)}$ as follows:

- for $p \in \mathcal{P}$, if $p \in \tau(i)$ then $p \in \pi(i)$, and if $\neg p \in \tau(i)$ then $p \notin \pi(i)$
- if $\psi_1 \wedge \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ and $\psi_2 \in \tau(i)$
- if $\psi_1 \vee \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ or $\psi_2 \in \tau(i)$

Labeling rules

$$\begin{aligned} \varphi \mathcal{U} \psi &\iff \psi \vee (\varphi \wedge \bigcirc(\varphi \mathcal{U} \psi)) \\ \varphi \mathcal{R} \psi &\iff \psi \wedge (\varphi \vee \bigcirc(\varphi \mathcal{R} \psi)) \end{aligned}$$

- if $\bigcirc \psi \in \tau(i)$ then $\psi \in \tau(i+1)$
- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$ then either $\psi_2 \in \tau(i)$, or $\psi_1 \in \tau(i)$ and $\psi_1 \mathcal{U} \psi_2 \in \tau(i+1)$
- if $\psi_1 \mathcal{R} \psi_2 \in \tau(i)$ then $\psi_2 \in \tau(i)$ and either $\psi_1 \in \tau(i)$ or $\psi_1 \mathcal{R} \psi_2 \in \tau(i+1)$

Interpreting labelings

A sequence π satisfies a formula φ if one can find a labeling τ satisfying:

- the labeling rules above
- $\varphi \in \tau(0)$, and
- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$, then for some $j \geq i$, $\psi_2 \in \tau(j)$ (the eventuality condition)

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

The automaton A_φ is the set of labeling rules + the eventuality condition(s) !

- $\Sigma = 2^{\mathcal{P}}$ is the alphabet
- $S \subseteq 2^{Cl(\varphi)}$, such that, for all $s \in S$:
  - $\varphi_1 \wedge \varphi_2 \in s \Rightarrow \varphi_1 \in s$ and $\varphi_2 \in s$
  - $\varphi_1 \vee \varphi_2 \in s \Rightarrow \varphi_1 \in s$ or $\varphi_2 \in s$
- $I = \{s \in S \mid \varphi \in s\}$,
- $(s, \alpha, t) \in T$ iff:
  - for all $p \in \mathcal{P}$, $p \in s \Rightarrow p \in \alpha$, and $\neg p \in s \Rightarrow p \notin \alpha$,
  - $\bigcirc\psi \in s \Rightarrow \psi \in t$,
  - $\psi_1 \mathcal{U} \psi_2 \in s \Rightarrow \psi_2 \in s$ or $[\psi_1 \in s$ and $\psi_1 \mathcal{U} \psi_2 \in t]$
  - $\psi_1 \mathcal{R} \psi_2 \in s \Rightarrow \psi_2 \in s$ and $[\psi_1 \in s$ or $\psi_1 \mathcal{R} \psi_2 \in t]$

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

- for each eventuality $\phi\mathcal{U}\psi \in Cl(\varphi)$, the transition relation ensures that this will appear until the first occurrence of ψ

- it is sufficient to ensure that, for each $\phi\mathcal{U}\psi \in Cl(\varphi)$, one goes infinitely often either through a state in which this does not appear, or through a state in which both $\phi\mathcal{U}\psi$ and ψ appear

- let $\phi_1\mathcal{U}\psi_1, \ldots \phi_n\mathcal{U}\psi_n$ be the "until" subformulae of φ

$\mathcal{F} = \{F_1, \ldots, F_n\}$, where:

$$F_i = \{s \in S \mid \phi_i\mathcal{U}\psi_i \in s \text{ and } \psi_i \in s \text{ or } \phi_i\mathcal{U}\psi_i \notin s\}$$

for all $1 \leq i \leq n$.