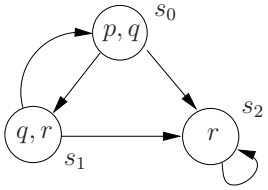
Session 3
Temporal Logics LTL

Contents

1 Linear Time

Model Checking



- Given a model M (*transition system*),
- a (initial) state s and a formula φ of a temporal logic (*specification*),
- the aim is
- $\mathcal{M}, s \models \varphi$
- i.e. φ is satisfied in the state s of $\mathcal{M}$
- a *model checker* is a program that decides this problem (i.e. either answers yes or no).

Examples of Properties

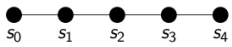
Propositional variables can be identified with states of reactive systems.

- It is impossible to get to a state where `started` holds, but `ready` does not hold.
- For any state, if `request` occurs, then it will eventually be acknowledged (`ack`).
- A certain process is `enabled` infinitely often on every execution.
- A process will eventually be permanently `deadlock`ed.

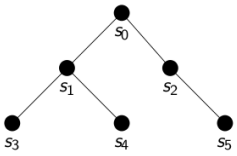
- If a process is `enabled` infinitely often, then it `runs` infinitely often.
- From any state it is possible to get to a `restart` state.

Time

Linear-Time: Time is a set of paths and a path is a sequence of states



Branching-time: Time is represented by a computation tree where the root is the present moment and each path from the root is an execution (sequence of states).



Computation Tree

Path/Execution

$T = (S, Act, \longrightarrow, AP, I, L)$ is a transition system, a path is a infinity sequence of states $s_1, s_2, s_3, \ldots$ in S, such that $i \geq 1$, $s_i \in Pre(s_{i+1})$. A path π is represented by

$$\pi = s_1 \longrightarrow s_2 \longrightarrow \cdots .$$

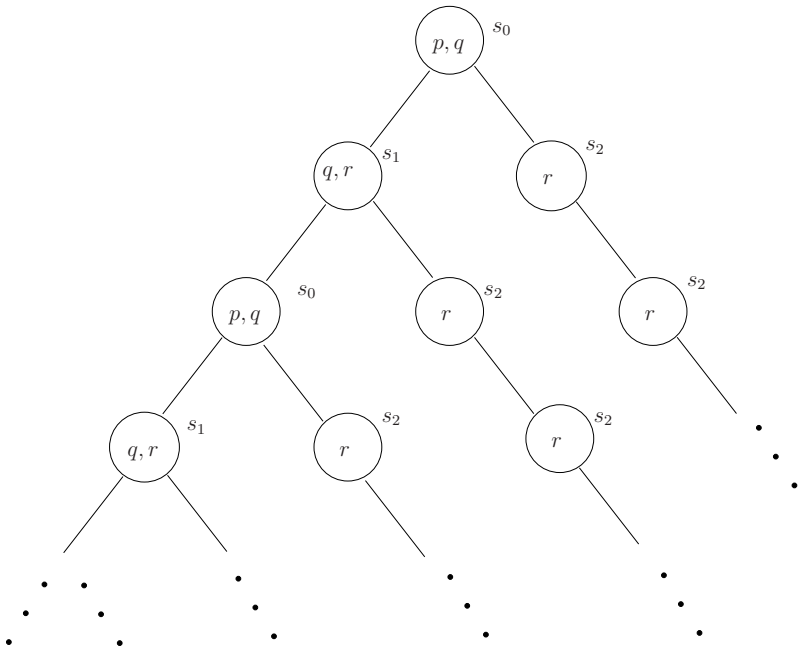
or

$$\pi = s_1 s_2 \cdots .$$

Given a path π,

- $\pi[i] = s_i$ (i.e. the ith state in the path)
- $\pi[i \ldots]$ is the suffix that starts at s_i.

The set of paths from a state s can be seen as an infinite *computation tree*.



Temporal Logics

Properties can be specified using first-order logic but it is not efficient.

Temporal logics

- are extensions of propositional and first-order logics
- use modalities to refer to the behaviour of the executions of reactive systems
- linear(-time) temporal logic (LTL) allows quantification over a path (path formulae)
- *computation tree logic* (CTL) allows quantification on the paths from a given state but in a restricted way (state formulae)
- logic CTL* allows unbounded quantification over path formulae. Includes LTL and CTL.

2 Linear-time Temporal Logic, LTL

Linear-time Temporal Logic, LTL

AP, set of propositional variables $p, q, r, s, \ldots$

$$\begin{array}{rcl} \varphi & ::= & \mathsf{false} \mid \mathsf{true} \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \\ & & (\mathrm{X}\varphi) \mid (\mathrm{F}\varphi) \mid (\mathrm{G}\varphi) \mid (\varphi \mathrm{U} \varphi) \mid (\varphi \mathrm{W} \varphi) \mid (\varphi \mathrm{R} \varphi) \end{array}$$

$\mathbf{X}\varphi$ φ holds in the *neXt* state (or $\bigcirc\varphi$)

$\mathbf{F}\varphi$ φ holds in some Future state (or $\Diamond\varphi$, *eventually*)

$\mathbf{G}\varphi$ φ holds Globally in every state (or $\Box\varphi$, always)

$\varphi\mathbf{U}\psi$ φ holds Until ψ

$\varphi\mathbf{W}\psi$ φ holds Until ψ or always (*Weak until*)

$\varphi\mathbf{R}\psi$ φ *Releases* ψ (ψ holds until φ or *always*)

Linear-time Temporal Logic, LTL

$$\begin{array}{l} (((\mathrm{F}p) \wedge (\mathrm{G}p)) \rightarrow (p \mathrm{W} r)) \\ ((\mathrm{G}(\mathrm{F}p)) \rightarrow (\mathrm{F}(q \vee p))) \\ (p \mathrm{W} (q \mathrm{W} r)) \end{array}$$

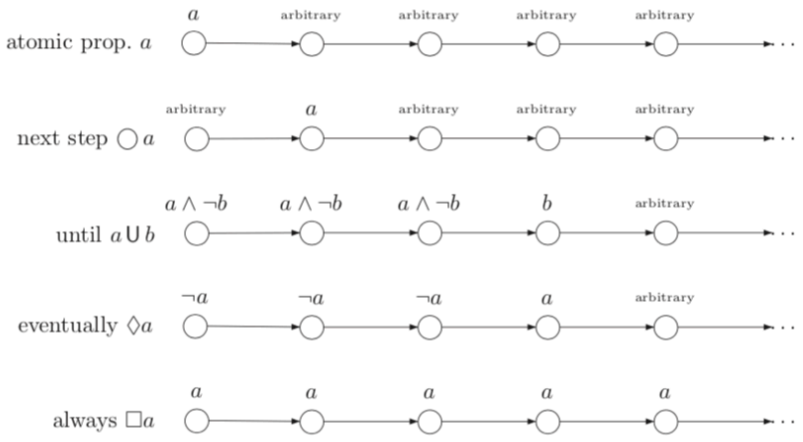
Binding Priorities

- Unary connectives ($\neg$, X, F,G) bind most tightly
- Next the connectives U, W and R.
- Then propositional connectives $\wedge$ and $\vee$.
- Next, the connective $\rightarrow$.

$$\begin{array}{l} \mathrm{F}p \wedge \mathrm{G}p \rightarrow p \mathrm{W} r \\ \mathrm{GF}p \rightarrow \mathrm{F}(q \vee p) \\ p \mathrm{W} (q \mathrm{W} r) \end{array}$$

Semantics of LTL

$\bigcirc$ is X; $\Diamond$ is F e $\Box$ is G



Semantics of LTL

Satisfiability

Given $\mathcal{M} = (S, \longrightarrow, L)$, a formula φ and a path $\pi = s_1 \longrightarrow s_2 \longrightarrow \cdots$ in $\mathcal{M}$, the satisfaction relation $\models$ (i.e. $\pi \models \varphi$) is defined inductively in the structure of φ as follows:

1. $\pi \models \mathsf{true}$
2. $\pi \not\models \mathsf{false}$
3. $\pi \models p$ iff $p \in L(s_1)$
4. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$
5. $\pi \models \varphi \wedge \psi$ iff $\pi \models \varphi$ and $\pi \models \psi$
6. $\pi \models \varphi \vee \psi$ iff $\pi \models \varphi$ or $\pi \models \psi$
7. $\pi \models \varphi \rightarrow \psi$ iff whenever $\pi \models \varphi$ then $\pi \models \psi$
8. $\pi \models \mathrm{X}\varphi$ iff $\pi[2\ldots] \models \varphi$
9. $\pi \models \mathrm{G}\varphi$ iff $\forall i \geq 1, \pi[i\ldots] \models \varphi$
10. $\pi \models \mathrm{F}\varphi$ iff $\exists i \geq 1, \pi[i\ldots] \models \varphi$
11. $\pi \models \varphi \mathrm{U} \psi$ iff $\exists i \geq 1, \pi[i\ldots] \models \psi$ and $\forall\, 1 \leq j < i$, $\pi[j\ldots] \models \varphi$
12. $\pi \models \varphi \mathrm{W} \psi$ iff or $\exists i \geq 1, \pi[i\ldots] \models \psi$ e $\forall\, 1 \leq j < i$, $\pi[j\ldots] \models \varphi$ or $\forall k \geq 1$, $\pi[k\ldots] \models \varphi$

13. $\pi \models \varphi \mathrm{R} \psi$ iff or $\exists i \geq 1, \pi[i \ldots] \models \varphi$ and $\forall 1 \leq j \leq i, \pi[j \ldots] \models \psi$ or $\forall k \geq 1, \pi[k \ldots] \models \psi$

Semantics of LTL

GF and FG

- $\mathrm{GF}\varphi$ means infinitely often
- $\mathrm{FG}\varphi$ means *from a certain point always*
- $\pi \models \mathrm{GF}\varphi$ iff $\overset{\infty}{\exists}\, j.\pi[j \ldots] \models \varphi$
- $\pi \models \mathrm{FG}\varphi$ iff $\overset{\infty}{\forall}\, j.\pi[j \ldots] \models \varphi$

where

$$\overset{\infty}{\exists}\, j \equiv \forall k \geq 1 \exists j \geq k$$

and

$$\overset{\infty}{\forall}\, j \equiv \exists k_0 \geq 1 \forall j \geq k_0$$

Examples

- *Safety*: Mutual exclusion $\mathrm{G}(\neg c_1 \vee \neg c_2)$
- *Liveness* : P_1 and P_2 access the critical section infinitely often $\mathrm{GF}c_1 \wedge \mathrm{GF}c_2$
- *Starvation freedom*: $(\mathrm{GF}w_1 \rightarrow \mathrm{GF}c_1) \wedge (\mathrm{GF}w_2 \rightarrow \mathrm{GF}c_2)$

Satisfiability in a State, LTL

Let $\mathcal{M} = (S, \longrightarrow, L)$, $s \in S$ and φ an LTL formula. We write

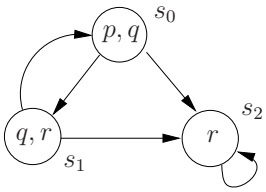
$$\mathcal{M}, s \models \varphi,$$

if *for all* paths π starting in s ($\pi \in Paths(s)$), we have $\pi \models \varphi$.

Exercice 3.1. *Consider the system* $\mathcal{M} = (S = \{s_0, s_1, s_2\}, \{s_0 \longrightarrow s_1, s_0 \longrightarrow s_2, s_1 \longrightarrow s_2, s_1 \longrightarrow s_0, s_2 \longrightarrow s_2\}, L(s_0) = \{p, q\}, L(s_1) = \{q, r\}, L(s_2) = \{r\})$. *Determine which relations are true*

1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models Xr$
3. $\mathcal{M}, s_0 \models X(q \wedge r)$

4. $\mathcal{M}, s_0 \models G\neg(p \wedge r)$
5. $\mathcal{M}, s_0 \models GFp$
6. $\mathcal{M}, s_0 \models GFp \rightarrow GFr$



Solution of Exercise 3.1

1. For all $\pi \in Paths(s_0)$, $\pi \models p \wedge q$ iff $p \in L(s_0)$ and $q \in L(s_0)$, which is true thus $\mathcal{M}, s_0 \models p \wedge q$.
2. For all $\pi \in Paths(s_0)$, $\pi \models \mathrm{X}r$ iff $r \in L(\pi[2])$, which is true as $\pi[2]$ is either s_1 or s_2. Thus $\mathcal{M}, s_0 \models \mathrm{X}r$.
3. Considering the previous answer, $\mathcal{M}, s_0 \not\models \mathrm{X}(q \wedge r)$ as $q \notin L(s_2)$.
4. As all states are reachable from s_0, we need to check that in all states s either $p \notin L(s)$ or $r \notin L(s)$, which is true. Thus, $\mathcal{M}, s_0 \models \mathrm{G}\neg(p \wedge r)$.
5. Using the notation introduced before we have to proof that $\overset{\infty}{\exists} j.\pi[j \ldots] \models p$, for all $\pi \in Paths(s_0)$. But for $\pi = s_0 s_2 s_2 s_2 \cdots$ that is not true as for $j > 1$ $\pi[j] = s_2$ and $p \notin L(s_2)$. For this π, $\pi \not\models \mathrm{GF}p$, and thus $\mathcal{M}, s_0 \not\models \mathrm{GF}p$.
6. We need to proof that for all $\pi \in Paths(s_0)$, $\pi \models \mathrm{GF}p \rightarrow \mathrm{GF}r$. For $\pi = s_0 s_2 s_2 s_2 \ldots$, as $\pi \not\models \mathrm{GF}p$ we have $\pi \models \mathrm{GF}p \rightarrow \mathrm{GF}r$. The same is true for $\pi = (s_0 s_1)^* s_2 s_2 \cdots$ (where * means any finite prefix $\underbrace{s_0 s_1 \cdots s_0 s_1}_{n}$, $n \geq 0$). Finally for $\pi = s_0 s_1 s_0 s_1 \cdots$, $\pi \models \mathrm{GF}p$, but also $\pi \models \mathrm{GF}r$ (for all $k \geq 1$, $\pi[k \ldots] \models \mathrm{F}r$ i.e. exists $j \geq k$ such that $\pi[j] = s_2$ and $\pi[j \ldots] \models r$)

Equivalence of LTL Formulas

Definition 1. Two formulas are semantically equivalent $\varphi_1 \equiv \varphi_2$, if for all models and paths π, $\pi \models \varphi_1$ iff π $\pi \models \varphi_2$

Exercice 3.2. *Show that*

$$\begin{aligned} F(\varphi \vee \psi) &\equiv F\varphi \vee F\psi \\ F(\varphi \wedge \psi) &\not\equiv F\varphi \wedge F\psi \\ G(\varphi \wedge \psi) &\equiv G\varphi \wedge G\psi \\ G(\varphi \vee \psi) &\not\equiv G\varphi \vee G\psi \end{aligned}$$

Solution of Exercise 3.2

Let $\mathcal{M}$ be a model and $\pi \in Paths(\mathcal{M}$. Then $\pi \models \mathrm{F}(\varphi \vee \psi$ iff exists $j \geq 1$ such that $\pi[j \ldots] \models \varphi \vee \psi$, i.e. $\pi[j \ldots] \models \varphi$ or $\pi[j \ldots] \models \psi$ iff exists $j \geq 1$ such that $\pi[j \ldots] \models \varphi$ or exists $j \geq 1$ such that $\pi[j \ldots] \models \psi$ iff $\pi \models \mathrm{F}\varphi$ or $\pi \models \mathrm{F}\psi$. Now to proof that $\mathrm{F}(\varphi \wedge \psi) \not\equiv \mathrm{F}\varphi \wedge \mathrm{F}\psi$, consider the model of Exercise 3.1 and $\pi = s_0 s_1 s_0 s_1 \cdots$. Then $\pi \models \mathrm{F}p$ and $\pi \models \mathrm{F}r$, but $\pi \not\models \mathrm{F}(p \wedge r)$. In a similar way one proves the two remaining equivalences.

Equivalence of LTL Formulas

Teorema 3.1. *We have*

$$\begin{array}{rcl}
\neg(\varphi \wedge \psi) & \equiv & \neg\varphi \vee \neg\psi \\
\neg(\varphi \vee \psi) & \equiv & \neg\varphi \wedge \neg\psi \\
\neg G\varphi & \equiv & F\neg\varphi \\
\neg F\varphi & \equiv & G\neg\varphi \\
\neg X\varphi & \equiv & X\neg\varphi \\
\neg(\varphi U\psi) & \equiv & \neg\varphi R\neg\psi \\
\neg(\varphi R\psi) & \equiv & \neg\varphi U\neg\psi \\
X(\varphi U\psi) & \equiv & (X\varphi)\,U(X\psi) \\
F(\varphi \vee \psi) & \equiv & F\varphi \vee F\psi \\
G(\varphi \wedge \psi) & \equiv & G\varphi \wedge G\psi
\end{array}$$

$\neg(\varphi \mathbf{U} \psi) \equiv \neg\varphi \mathbf{R} \neg\psi$

If $\pi \not\models \varphi \mathrm{U} \psi$ then

- or $\forall\, i$, $\pi[i \ldots] \models \neg\psi$
- or $\exists i \geq 0, \pi[i \ldots] \models \neg\varphi$ and $\forall\, 0 \leq j \leq i$, $\pi[j \ldots] \models \neg\psi$
- that is $\pi \models \neg\varphi \mathrm{R} \neg\psi$

More Equivalences

$$
\begin{aligned}
\mathrm{F}\varphi &\equiv \mathsf{true}\mathrm{U}\varphi \\
\mathrm{G}\varphi &\equiv \mathsf{false}\mathrm{R}\varphi \\
\varphi\mathrm{U}\psi &\equiv \varphi\mathrm{W}\psi \wedge \mathrm{F}\psi \\
\varphi\mathrm{W}\psi &\equiv \varphi\mathrm{U}\psi \vee \mathrm{G}\varphi \\
\varphi\mathrm{W}\psi &\equiv \psi\mathrm{R}(\varphi \vee \psi) \\
\varphi\mathrm{R}\psi &\equiv \psi\mathrm{W}(\varphi \wedge \psi) \\
\varphi\mathrm{U}\psi &\equiv \psi \vee (\varphi \wedge \mathrm{X}(\varphi\mathrm{U}\psi)) \\
\mathrm{F}\varphi &\equiv \varphi \vee \mathrm{XF}\varphi \\
\mathrm{G}\varphi &\equiv \varphi \wedge \mathrm{XG}\varphi
\end{aligned}
$$

Exercice 3.3. *Proof the above equivalences.*

Complete sets of connectives for LTL

A set of connectives is complete if the remaining can be defined from the elements of that set.

Teorema 3.2. *The following sets of temporal connectives are complete for LTL:* $\{U, X\}$, $\{R, X\}$ *and* $\{W, X\}$.

Exercice 3.4. *Show that* $\{U, X\}$ *is complete for LTL.*

Solution of Exercise 3.4

We need to show how the connectives $\mathrm{F}, \mathrm{G}, \mathrm{W}$ adn R can be written using $\{\mathrm{U}, \mathrm{X}\}$ and proosicional connectives.

$$
\begin{aligned}
\mathrm{F}\varphi &\equiv \mathsf{true}\mathrm{U}\varphi \\
\mathrm{G}\varphi &\equiv \neg\mathsf{true}\mathrm{U}\neg\varphi \\
\varphi\mathrm{W}\psi &\equiv \varphi\mathrm{U}\psi \vee \mathrm{G}\varphi \text{ and use the previous one} \\
\varphi\mathrm{R}\psi &\equiv \psi\mathrm{W}(\varphi \wedge \psi) \text{ and use the previous one}
\end{aligned}
$$

Property Specification

- Is is impossible to get to a state where `started` holds, but `ready` does not hold. $G\neg($`started` $\wedge$ $\neg$`ready`$)$
- For any state, if `request` occurs, then it will eventually be acknowledged, `ack`. $G($`request` $\rightarrow F$`ack`$)$

- A certain process is `enabled` infinitely often on every execution. GF`enabled`
- A process will eventually be permanently `deadlock`ed. FG`deadlock`
- If a process is `enabled` infinitely often, then it `run`s infinitely often GF`enabled` $\rightarrow GF$`run`
- From any state it is possible to get to a `restart` state. *It is not possible to express in LTL*