

Lógica LTL

Aula 9

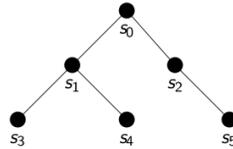
1 Noção de Tempo

Tempo

Linear: O tempo é um conjunto de caminhos, onde um caminho é uma sequência de estados.



Ramificado: O tempo é representado em árvore com raiz no momento presente.



Árvore Infinita de Computação

Caminho (de Computação)

$T = (S, Act, \longrightarrow, AP, I, L)$ um sistema de transição, um caminho é uma sequência infinita de estados s_1, s_2, s_3, \dots em S , tal que para todo $i \geq 1$, $s_i \in Pre(s_{i+1})$.

O conjunto de caminhos a partir dum estado s pode ser visto como uma **árvore infinita de computação**.

- $X\varphi$ φ verifica-se no estado seguinte (*neXt*) (ou $\bigcirc\varphi$)
- $F\varphi$ φ verifica-se **nalgum** estado futuro (ou $\diamond\varphi$, *eventually*)
- $G\varphi$ φ verifica-se **em todos** os estados futuros (*Global*) (ou $\square\varphi$, sempre)
- $\varphi U\psi$ φ verifica-se **até** ψ se verificar (*Until*)
- $\varphi W\psi$ φ verifica-se **até** ψ se verificar ou **sempre** (*Weak until*)
- $\varphi R\psi$ ψ verifica-se **até** φ se verificar (inclusivél) ou **sempre** (*Release*)

Lógica Temporal Linear, LTL

$$\begin{aligned} &(((Fp) \wedge (Gp)) \rightarrow (pWr)) \\ &((G(Fp)) \rightarrow (F(q \vee p))) \\ &(pW(qWr)) \end{aligned}$$

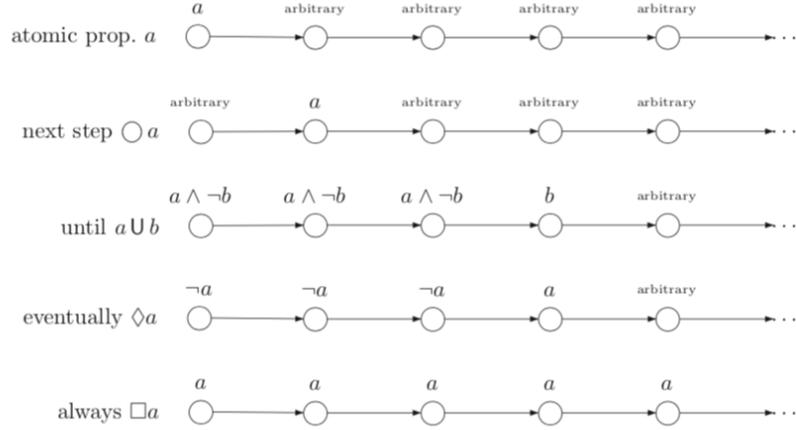
Convensão de prioridades (omissão de parêntesis)

- As conectivas unárias (\neg , X, F,G) têm prioridade mais alta
- Depois as conectivas U, W e R.
- Depois as conectivas \wedge e \vee .
- Depois a conectiva \rightarrow .

$$\begin{aligned} &Fp \wedge Gp \rightarrow pWr \\ &GFp \rightarrow F(q \vee p) \\ &pW(qWr) \end{aligned}$$

Semântica do LTL

\bigcirc é X; \diamond é F e \square é G



GF e FG

- $GF\varphi$ significa *um número infinito de vezes*
- $FG\varphi$ significa *a partir de um certo momento, para sempre*

Exemplos

- Exclusão mútua (*segurança*): $\{A_0A_1A_2\dots \mid \{c_1, c_2\} \not\subseteq A_i, i \geq 0, A_i \subseteq AP\}$

$$G(\neg c_1 \vee \neg c_2)$$

- P_i acede à secção crítica um n.i.d.v (*vivacidade/liveness*):

$$\{A_0A_1A_2\dots \mid (\exists^\infty j. c_1 \in A_j) \wedge (\exists^\infty j. c_2 \in A_j)\}$$

$$GFc_1 \wedge GFc_2$$

- *starvation freedom* (vivacidade):

$$\{A_0A_1A_2\dots \mid \forall i \in \{1, 2\}, (\exists^\infty j. w_i \in A_j) \rightarrow (\exists^\infty j. c_i \in A_j)\}$$

$$(GFw_1 \rightarrow GFc_1) \wedge (GFw_2 \rightarrow GFc_2)$$

Semântica do LTL

Seja $T = (S, Act, \longrightarrow, I, AP, L)$ e um traço

$$\sigma = A_0 A_1 A_2 \dots,$$

A propriedade LT induzida por φ é

$$Words(\varphi) = \{ \sigma \in (2^{AP})^\omega \mid \sigma \models \varphi \}$$

onde a relação de satisfabilidade $\sigma \models \varphi$ define-se indutivamente por:

Satisfazibilidade

1. $\sigma \models \text{true}$
2. $\sigma \not\models \text{false}$
3. $\sigma \models p$ sse $p \in A_0$
4. $\sigma \models \neg\varphi$ sse $\sigma \not\models \varphi$
5. $\sigma \models \varphi \wedge \psi$ sse $\sigma \models \varphi$ e $\sigma \models \psi$
6. $\sigma \models \varphi \vee \psi$ sse $\sigma \models \varphi$ ou $\sigma \models \psi$
7. $\sigma \models \varphi \rightarrow \psi$ sse sempre que $\sigma \models \varphi$ então $\sigma \models \psi$
8. $\sigma \models X\varphi$ sse $\sigma[1\dots] \models \varphi$
9. $\sigma \models G\varphi$ sse $\forall i \geq 0, \sigma[i\dots] \models \varphi$
10. $\sigma \models F\varphi$ sse $\exists i \geq 0, \sigma[i\dots] \models \varphi$
11. $\sigma \models \varphi U \psi$ sse $\exists i \geq 0, \sigma[i\dots] \models \psi$ e $\forall 0 \leq j < i, \sigma[j\dots] \models \varphi$
12. $\sigma \models \varphi W \psi$ sse ou $\exists i \geq 0, \sigma[i\dots] \models \psi$ e $\forall 0 \leq j < i, \sigma[j\dots] \models \varphi$ ou $\forall k \geq 0, \sigma[k\dots] \models \varphi$
13. $\sigma \models \varphi R \psi$ sse ou $\exists i \geq 0, \sigma[i\dots] \models \varphi$ e $\forall 0 \leq j \leq i, \sigma[j\dots] \models \psi$ ou $\forall k \geq 0, \sigma[k\dots] \models \psi$

E também...

- $\sigma \models GF\varphi$ sse $\overset{\infty}{\exists} j. \sigma[j\dots] \models \varphi$
- $\sigma \models FG\varphi$ sse $\overset{\infty}{\forall} j. \sigma[j\dots] \models \varphi$

onde

$$\overset{\infty}{\exists} j \equiv \forall k \geq 0 \exists j \geq k$$

e

$$\overset{\infty}{\forall} j \equiv \exists k_0 \geq 0 \forall j \geq k_0$$

Traços (revisão)

- $T = (S, Act, \longrightarrow, I, AP, L)$ sistema de transições

$$\begin{aligned}\rho &= s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \dots \\ \pi &= s_0 s_1 s_2 \dots \\ trace(\pi) &= L(s_0)L(s_1)L(s_2)\dots\end{aligned}$$

- Π conjunto de caminhos, $trace(\Pi) = \{trace(\pi) \mid \pi \in \Pi\}$.
- $trace(s) = \{trace(\pi) \mid s = first(\pi)\}$
- $Traces(s) = trace(Paths(s))$
- $Traces(T) = \bigcup_{s \in I} Traces(s)$

Semântica do LTL para caminhos, estados e sistemas

Seja $T = (S, Act, \longrightarrow, I, AP, L)$, π um fragmento de caminho infinito e $s \in S$.

- $\pi \models \varphi$ sse $trace(\pi) \models \varphi$
- $s \models \varphi$ sse *para todos* os caminhos de computação π começando em s , se tem $\pi \models \varphi$, i.e.

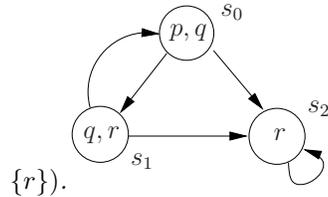
$$\forall \pi \in Paths(s), \pi \models \varphi.$$

- T satisfaz φ sse $Traces(T) \subseteq Words(\varphi)$, ou seja, sse

$$s_0 \models \varphi, \forall s_0 \in I.$$

Semântica do LTL

Exercício 9.1. Considera o sistema $T = (S = \{s_0, s_1, s_2\}, \{s_0 \longrightarrow s_1, s_0 \longrightarrow s_2, s_1 \longrightarrow s_2, s_1 \longrightarrow s_0, s_2 \longrightarrow s_2\}, L(s_0) = \{p, q\}, L(s_1) = \{q, r\}, L(s_2) = \{r\})$.

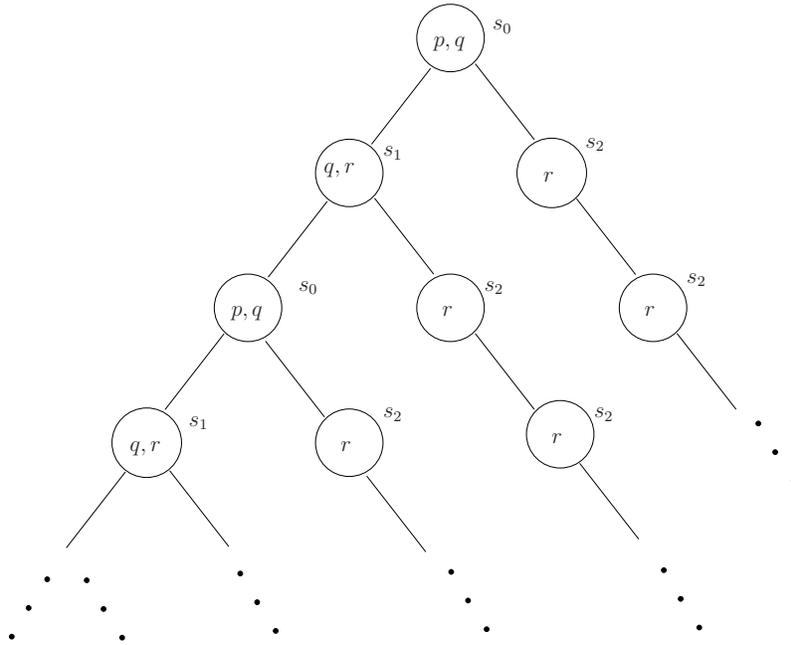


Determina quais destas relações são verdadeiras:

1. $s_0 \models p \wedge q$
2. $s_0 \models Xr$

3. $s_0 \models X(q \wedge r)$
4. $s_0 \models G\neg(p \wedge r)$
5. $s_0 \models GFp$
6. $s_0 \models GFp \rightarrow GFr$

◇



Algumas especificações práticas (padrões)

As variáveis proposicionais correspondem a estados de uma sistema reactivo real.

- Não é possível chegar a um estado em que **started** se verifica e **ready** não se verifica: $G\neg(\text{started} \wedge \neg\text{ready})$
- Para qualquer estado, se **request** se verifica, então no futuro também se irá verificar **ack**: $G(\text{request} \rightarrow F\text{ack})$
- Um processo é activado (**enabled**) um número infinito de vezes em todos os caminhos de computação: $GF\text{enabled}$
- Um processo irá ficar permanentemente em **deadlock**: $FG\text{deadlock}$
- Se um processo é activado (**enabled**) um número infinito de vezes, então ele executa (**run**) um número infinito de vezes: $GF\text{enabled} \rightarrow GF\text{run}$

- Para todos os estados, existe um caminho para um estado que satisfaz **restart**: **Não se pode exprimir em LTL!**

Em LTL não se pode exprimir a existência de um caminho de um estado com uma dada propriedade.

Exercício 9.2. *Considera o sistema $\mathcal{M} = (S = \{q_1, q_2, q_3, q_4\}, \{q_1 \rightarrow q_2, q_2 \rightarrow q_3, q_3 \rightarrow q_1, q_3 \rightarrow q_2, q_3 \rightarrow q_4, q_4 \rightarrow q_3\}, L(q_1) = \{\}, L(q_2) = \{b\}, L(q_3) = \{a\}, L(q_4) = \{a, b\})$.*

Para cada uma das fórmulas seguintes:

- Ga ;
- aUb ;
- $aUX(a \wedge \neg b)$;
- $X\neg b \wedge G(\neg a \vee \neg b)$;
- $X(a \wedge b) \wedge F(\neg a \wedge \neg b)$;

- *encontra um caminho a partir do estado q_3 que satisfaz φ ;*
- *determina se $q_3 \models \varphi$.*

◇

3 Especificações

Equivalência semântica

Equivalência de fórmulas

Duas fórmulas em LTL são semanticamente equivalentes, $\varphi \equiv \psi$, sse $Words(\varphi_1) = Words(\varphi_2)$

Exercício

Mostra que

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

Equivalência semântica

Teorema 9.1. *Temos as seguintes equivalências:*

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

Mais equivalências

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

Conjuntos completos de conectivas

Um conjunto de conectivas é completo se as restantes se podem obter em termos dos seus elementos

Teorema 9.2. *Considerando as conectivas temporais, os seguintes conjuntos são completos: $\{U, X\}$, $\{R, X\}$ e $\{W, X\}$.*

Exercício

Mostre que $\{U, X\}$ é completo para LTL.

Exercícios

Exercício 9.3. *Demonstra a veracidade ou falsidade das afirmações seguintes com uma prova ou um contra-exemplo.*

a) $G(\varphi \vee \psi) \equiv G(\varphi) \vee G(\psi)$

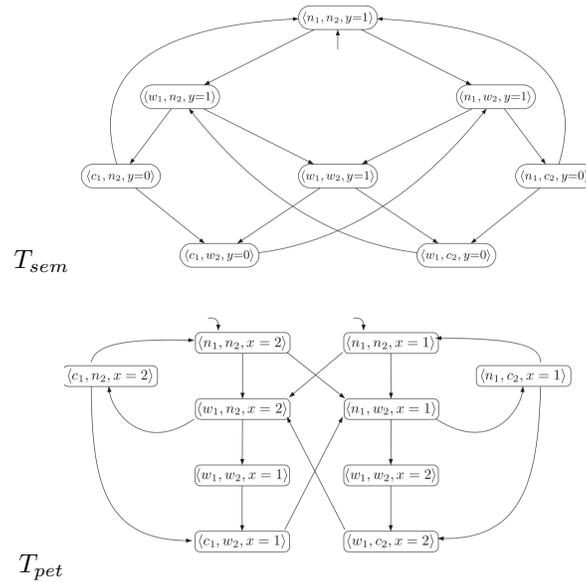
b) $G(\varphi \wedge \psi) \equiv G(\varphi) \wedge G(\psi)$

◇

Exercício 9.4. *Indica uma fórmula em LTL que é semânticamente equivalente a*

$$\neg G(c_1 \rightarrow c_1 W(\neg c_1 \wedge \neg c_1 W c_2))$$

mas não utiliza a conectiva W. ◇



Especificação das propriedades em LTL

Considera os sistemas para a exclusão mútua T_{sem} (com semáforo) e T_{pet} (de Peterson). E as propriedades

de Segurança Já vimos que $T_{sem} \models G\neg(c_1 \wedge c_2)$

de Vivacidade $T_{sem} \not\models G(w_1 \rightarrow Fc_1)$, mas $T_{pet} \models G(w_1 \rightarrow Fc_1)$

Non-blocking Cada estado em que n_i se verifica existe um sucessor tal que w_i se verifica. *Não se pode exprimir em LTL.*

Not strict sequencing Existe um caminho com dois estados que satisfazem c_1 , tal que nenhum estado entre eles verifica c_2 . *Isto não se pode exprimir*

directamente em LTL. Mas, usando o complementar: Todos os caminhos com um período em que c_1 é satisfeito, mas que termina, não podem ter c_1 antes de ter c_2 :

$$G(c_1 \rightarrow c_1 W(\neg c_1 \wedge \neg c_1 W c_2))$$

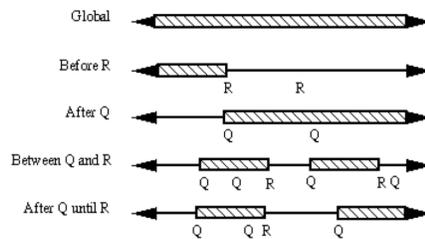
Complementando temos a propriedade pretendida, e que se verifica em T_{sem} . Porquê?

Especificações de Propriedades

- Ocorrência: existência ou não de certos estados numa dada região
 - Ausência
 - Universalidade
 - Existência
 - Existência limitada
- Ordem: relaciona pares de estados numa dada região
 - Precedência
 - Sequência (*)

Regiões

- Global
- Antes de r
- Depois de q
- Entre q e r
- Depois de q até r



Especificações em LTL

Ausência/Universalidade

$\varphi: \neg p / p$

Global	$G\varphi$
Antes de r	$Fr \rightarrow \varphi Ur$
Depois de q	$G(q \rightarrow G\varphi)$
Entre q e r	$G((q \wedge \neg r \wedge Fr) \rightarrow (\varphi Ur))$
Depois de q até r	$G(q \wedge \neg r \rightarrow (\varphi Wr))$

Existência

p passa a ser verdade

Global	Fp
Antes de r	$\neg r W(p \wedge \neg r)$
Depois de q	$G(\neg q) \vee F(q \wedge Fp)$
Entre q e r	$G(q \wedge \neg r \rightarrow (\neg r W(p \neg r)))$
Depois de q até r	$G(q \wedge \neg r \rightarrow (\neg r U(p \wedge \neg r)))$

Especificações em LTL

Existência limitada

Ex: transições para estados com p ocorrem no máximo 2 vezes

Global	$\neg p W(p W(\neg p W(p W G \neg p)))$
Antes de r	$Fr \rightarrow ((\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p U r))))))))$
Depois de q	$Fq \rightarrow (\neg q U(q \wedge \neg p W(p W(\neg p W(p W G \neg p))))$
Entre q e r	$G((q \wedge \neg r) \rightarrow ((\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p U r))))))))$
Depois de q até r	$G(q \rightarrow ((\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p \wedge \neg r) U(r \vee ((p \wedge \neg r) U(r \vee (\neg p W r) \vee G p))))))))$

Especificações em LTL

Precedência

s precede p

Global	$\neg pWs$
Antes de r	$Fr \rightarrow (\neg pU(s \vee r))$
Depois de q	$G(\neg q) \vee F(q \wedge (\neg pWs))$
Entre q e r	$G((q \wedge \neg r \wedge Fr) \rightarrow (\neg pU(s \vee r)))$
Depois de q até r	$G(q \wedge \neg r \rightarrow (\neg pW(s \vee r)))$

Justeza baseada em ações

- Dado $T = (S, Act, \longrightarrow, AP, I, L)$
- $Act(s) = \{ \alpha \in Act \mid \exists s' \in S. s \xrightarrow{\alpha} s' \}$, ações executáveis em s
- Seja $A \subseteq Act$
- $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$,
- ρ é *incondicionalmente* A -justo se

$$\exists^{\infty} j. \alpha_j \in A.$$

- ρ é *fortemente* A -justo se

$$(\exists^{\infty} j. Act(s_j) \cap A \neq \emptyset) \rightarrow \exists^{\infty} j. \alpha_j \in A.$$

- ρ é *fracamente* A -justo se

$$(\forall^{\infty} j. Act(s_j) \cap A \neq \emptyset) \rightarrow \exists^{\infty} j. \alpha_j \in A.$$

- onde $\forall^{\infty} j$ significa todos j excepto um número finito.

Restrições de Fairness/Justeza em LTL

Sejam Φ e Ψ duas fórmulas proposicionais sobre AP . Uma restrição de justeza

- *incondicional* é da forma $GF\Phi$ (ufair)
- *forte* é da forma $GF\Phi \rightarrow GF\Psi$ (sfair)
- *fraca* é da forma $FG\Phi \rightarrow GF\Psi$ (wfair)

uma assunção de justeza é uma conjunção de restrições de justeza

$$fair = ufair \wedge sfair \wedge wfair$$

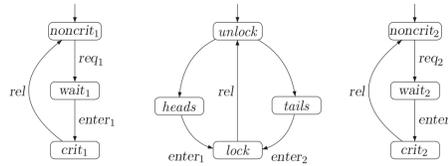
Satisfazibilidade para LTL com Fairness

$$\begin{aligned} FairPaths(s) &= \{\pi \in Paths(s) \mid \pi \models fair\} \\ FairTraces(s) &= \{trace(\pi) \mid \pi \in FairPaths(s)\} \end{aligned}$$

- $s \models_{fair} \varphi$ sse $\forall \pi \in FairPaths(s), \pi \models \varphi$
- $T \models_{fair} \varphi$ sse $\forall s_0 \in I. s_0 \models_{fair} \varphi$

Exclusão mútua com arbitro aleatório

O árbitro decide quem entra na região crítica lançando uma moeda ao ar, cujo resultado é simulado pelas ações *heads* (cara) e *tails* (coroa).



- Tem-se que $T_1 || Arbiter || T_2 \not\models GFc_1$. Porquê?
- Agora se $fair = GFheads \wedge GFtails$
- então teríamos $T_1 || Arbiter || T_2 \models_{fair} GFc_1$.
- em geral $T \models_{fair} \varphi$ sse $T \models (fair \rightarrow \varphi)$ (Prova!)