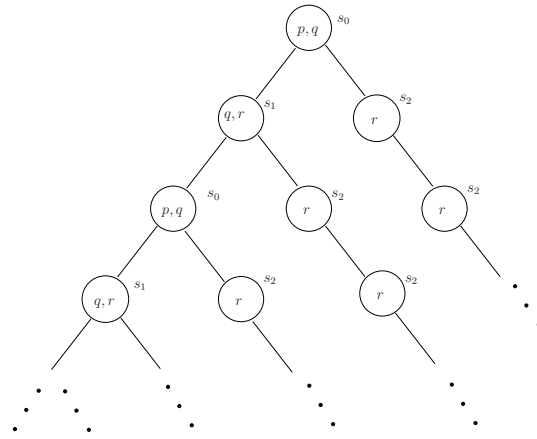


Lógica CTL

Aula 11

Lógica de tempo ramificado - CTL

A evolução de um sistema de transições corresponde a uma árvore de computação infinita. A lógica CTL (*Computation Tree Logic*) permite quantificação (existencial) sobre os caminhos dessa árvore.



Lógica de tempo ramificado - CTL

AP, conjunto de variáveis proposicionais p, q, r, s, \dots

Sintaxe

$$\varphi ::= \text{true} \mid \text{false} \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\text{AX}\varphi) \mid (\text{EX}\varphi) \mid (\text{AF}\varphi) \mid (\text{EF}\varphi) \mid (\text{AG}\varphi) \mid (\text{EG}\varphi) \mid \text{A}[\varphi\text{U}\varphi] \mid \text{E}[\varphi\text{U}\varphi]$$

Conectivas Temporais

A significa *ao longo de todos os caminhos*

E significa *ao longo de pelo menos um caminho*

F, G, X e **U** como no LTL

Lógica de tempo ramificado - CTL

As conectivas **A** e **E** só podem aparecer junto de uma das outras conectivas temporais.

Convensão de prioridades (omissão de parêntesis)

- As conectivas unárias (\neg , AX, EX, AF, EF, AG e EG) têm prioridade mais alta
- Depois as conectivas \wedge e \vee .
- Depois as conectivas \rightarrow , AU e EU (estas duas últimas são escritas em notação infix e prefixa simultaneamente)

Exemplos

$$\begin{aligned} &AG(p \rightarrow EGr) \\ &EFE[rUq] \\ &E[A[rUp]Uq] \\ &A[AX\neg pUE[EX(p \wedge q)U\neg p]] \end{aligned}$$

1 Semântica do CTL

Exemplos:

Exprime as propriedades seguintes através de fórmulas em CTL.

- Existe um estado atingível onde se verifica p . EFp
- A partir de todos os estados atingíveis, onde se verifica p , é possível manter p continuamente verdadeiro até chegar a um estado onde se verifica q . $AG(p \rightarrow EpUq)$
- Sempre que se chega a um estado onde se verifica p , é possível manter q verdadeiro para sempre. $(AG(p \wedge AGq))$
- Existe um estado atingível, a partir do qual se verifica p em todos os estados atingíveis. $EGAGp$

Sempre e Eventualmente

- $EF\varphi = EtrueU\varphi$, φ verifica-se potencialmente
- $AF\varphi = AtrueU\varphi$, φ verifica-se inevitavelmente
- $EG\varphi = \neg AF\neg\varphi$, φ verifica-se potencialmente sempre
- $AG\varphi = \neg EF\neg\varphi$, φ invariavelmente
- $AGAF\varphi$, φ verifica-se um n.i.d.v. em todos os caminhos.

Exemplos

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

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

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

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

$$AGAF_{c_1} \wedge AGAF_{c_2}$$

- *starvation freedom* (*vivacidade*):

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

$$(AGAF_{w_1} \rightarrow AGAF_{c_1}) \wedge (AGAF_{w_2} \rightarrow AGAF_{c_2})$$

Semântica do CTL

Satisfazibilidade

Dado um sistema de transições $T = (S, Act, \longrightarrow, AP, I, L)$, um estado $s \in S$, para $\pi \in Paths(s)$ seja $\pi = s_0s_1\dots$, $s_0 = s$ e $\pi[i] = s_i$ é o $i + 1$ estado em π .

Dada uma fórmula φ define-se a relação de satisfazibilidade $s \models \varphi$ indutivamente na estrutura de φ :

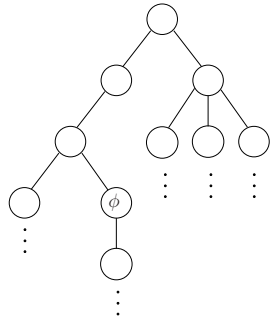
1. $s \models \text{true}$ e $s \not\models \text{false}$
2. $s \models p$ sse $p \in L(s)$
3. $s \models \neg\varphi$ sse $s \not\models \varphi$
4. $s \models \varphi \wedge \psi$ sse $s \models \varphi$ e $s \models \psi$
5. $s \models \varphi \vee \psi$ sse $s \models \varphi$ ou $s \models \psi$
6. $s \models \varphi \rightarrow \psi$ sse se $s \models \varphi$ então $s \models \psi$
7. $s \models AX\varphi$ sse para todo $\pi \in Paths(s)$, $\pi[1] \models \varphi$
8. $s \models EX\varphi$ sse existe $\pi \in Paths(s)$, $\pi[1] \models \varphi$

Semântica do CTL

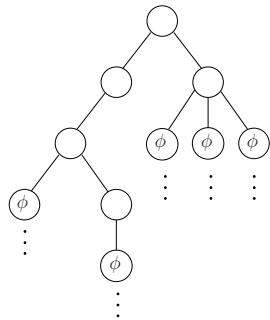
Satisfazibilidade (cont.)

9. $s \models \text{AG}\varphi$ sse para todos os caminhos $\pi \in \text{Paths}(s)$, se tem para todo $i \geq 0$ $\pi[i] \models \varphi$
10. $s \models \text{EG}\varphi$ sse existe um caminho $\pi \in \text{Paths}(s)$ tal que para todo $i \geq 0$, $\pi[i] \models \varphi$
11. $s \models \text{AF}\varphi$ sse para todos os caminhos $\pi \in \text{Paths}(s)$, existe $i \geq 0$ tal que $\pi[i] \models \varphi$
12. $s \models \text{EF}\varphi$ sse existe um caminho $\pi \in \text{Paths}(s)$ tal que existe $i \geq 0$ tal que $\pi[i] \models \varphi$
13. $\mathcal{M}, s \models \text{A}[\varphi_1 \text{U}\varphi_2]$ sse para todos os caminhos $\pi \in \text{Paths}(s)$, se tem que $\varphi_1 \text{U}\varphi_2$ é satisfeito, i.e. existe $i \geq 0$ $\pi[i] \models \varphi_2$, e para $0 \leq j < i$ $\pi[j] \models \varphi_1$
14. $s \models \text{E}[\varphi_1 \text{U}\varphi_2]$ sse existe um caminho $\pi \in \text{Paths}(s)$, tal que $\varphi_1 \text{U}\varphi_2$ é satisfeito, i.e. existe $i \geq 0$ $\pi[i] \models \varphi_2$, e para $0 \leq j < i$ $\pi[j] \models \varphi_1$.

Semântica do CTL

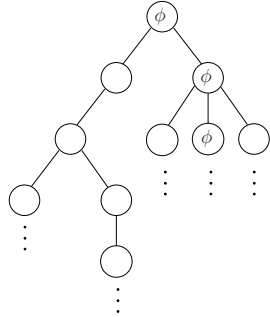


$\text{EF}\phi$

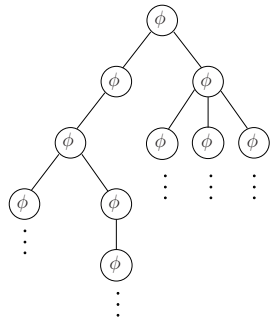


$\text{AF}\phi$

Semântica do CTL



EGφ



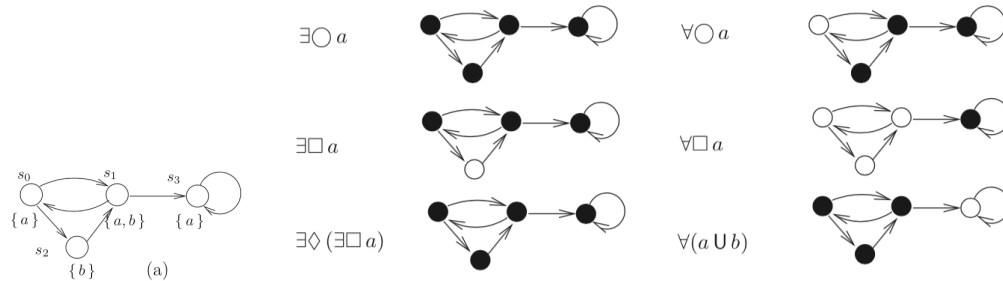
AGφ

Semântica do CTL para sistemas de transições

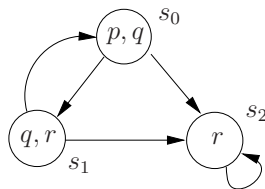
- Dado um sistema $T = (S, Act, \longrightarrow, AP, I, L)$ e uma fórmula φ de CTL,
- O conjunto de satisfação para φ é $Sat(\varphi) = \{ s \in S \mid s \models \varphi \}$
- T satisfaz φ i.e $T \models \varphi$ sse $\forall s_0 \in I, s_0 \models \varphi$ (i.e $I \subseteq Sat(\varphi)$)

Exemplo

Indica quais os estados em que as seguintes fórmulas são satisfeitas: EXa , AXa , EGa , AGa , $EFEGa$ e $A[aUb]$,



Exemplo



1. $s_0 \models p \wedge q$
2. $s_0 \models EXr$
3. $s_0 \models \neg AX(q \wedge r)$
4. $s_0 \models \neg EF(p \wedge r)$
5. $s_0 \models EGr$
6. $s_0 \models A[pUr]$

Especificação de algumas propriedades do CTL

Exercício 11.1. *Mostra que uma fórmula φ é satisfeita um número infinito de vezes ao longo de qualquer caminho a partir de s de um modelo \mathcal{M} sse e*

$$s \models AGAF\varphi$$

◇

Exemplos:

Exprime as propriedades seguintes através de fórmulas em CTL

- É possível atingir um estado onde se verifica `started` e onde `ready` é falso. ($EF(\text{started} \wedge \neg \text{ready})$)

- Em qualquer estado, se se verificar **trying**, então existe um caminho onde **critical** se verifica mais tarde (non-blocking). $(AG(trying \rightarrow EFcritical))$
- Um processo é **enabled** um número infinito de vezes ao longo de qualquer caminho. $(AG(AFenabled))$
- Ao longo de qualquer caminho, se **enabled** se verificar um número infinito de vezes, então **running** verifica-se um número infinito de vezes. Não é possível. Não é $(AGAFenabled \rightarrow AGAFrunning)$
- De qualquer estado é possível atingir um estado onde se verifica **restart**. $(AGEFrestart)$

Equivalência semântica

Equivalência de fórmulas

Duas fórmulas do CTL (sobre AP) são semanticamente equivalentes, $\varphi \equiv \psi$, se $Sat(\varphi) = Sat(\psi)$ para todos os sistemas de transições T sobre AP (i.e $T \models \varphi$ sse $T \models \psi$)

Temos as seguintes equivalências:

$$\begin{aligned} \neg AF\varphi &\equiv EG\neg\varphi \\ \neg EF\varphi &\equiv AG\neg\varphi \\ \neg AX\varphi &\equiv EX\neg\varphi \\ AF\varphi &\equiv A[\text{true}U\varphi] \\ EF\varphi &\equiv E[\text{true}U\varphi] \end{aligned}$$

Conjuntos completos de conectivas

Teorema 11.1. *Considerando as conectivas temporais, os seguintes conjuntos são completos: $\{AU, EU, EX\}$ e $\{EG, EU, EX\}$.*

Teorema 11.2. *Um conjunto de conectivas temporais do CTL é completo se contiver pelo menos um elemento do conjunto $\{AX, EX\}$, um do conjunto $\{EG, AF, AU\}$ e EU .*

Mais equivalências

$$\begin{aligned} AG\varphi &\equiv \varphi \wedge AXAG\varphi \\ EG\varphi &\equiv \varphi \wedge EXEG\varphi \\ AF\varphi &\equiv \varphi \vee AXAF\varphi \\ EF\varphi &\equiv \varphi \vee EXEF\varphi \end{aligned}$$

LTL e CTL

O CTL não é estritamente mais expressivo que o LTL. Por exemplo

$$Fp \rightarrow Fq$$

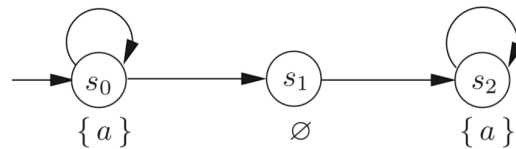
não se pode exprimir em CTL... O seu significado é

Todos os caminhos em que p é se verifica, também se verifica q .

Vê o que significa $AFp \rightarrow AFq$, ou $AG(p \rightarrow AFq)$.

LTL E CTL

$FG\varphi$ não é $AFAG\varphi$



$$s_0 \models FGa$$

mas

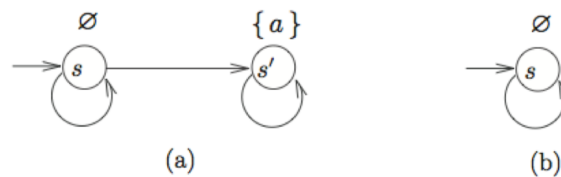
$$s_0 \not\models AFAGa$$

(verifica que para s_0' existe um estado (s_0) e $s_0 \not\models AGa$.)

LTL e CTL

Mas $AGEFa$ não se pode exprimir em LTL:

A partir de qualquer estado é possível atingir um estado em que a é verdade.



Não existe nenhuma formula φ em LTL equivalente. Porque se houvesse $M_a, s \models AGEFa$ e $M_b, s \not\models AGEFa$, mas ambos satisfazem φ . Verifica!

LTL e CTL

Analogamente, tem-se que $FXa \equiv XFa \equiv AXAFa$ mas

$$FXa \not\equiv AFAXa.$$

Teorema 11.3. *Seja ψ uma fórmula de CTL e φ a fórmula de LTL que se obtém eliminando os operadores A e E. Então ou*

$$\psi \equiv \varphi$$

ou não existe uma fórmula de LTL equivalente a ψ .

LTL versus CTL

Teorema 11.4. *a) Existem fórmulas LTL para as quais não existe fórmula CTL equivalente. Por exemplo, FGa.*

b) Existem fórmulas CTL para as quais não existe fórmula LTL equivalente. Por exemplo, AGEFa.

LTL versus CTL

<i>Aspect</i>	<i>Linear time</i>	<i>Branching time</i>
“behavior” in a state s	path-based: $trace(s)$	state-based: computation tree of s
temporal logic	LTL: path formulae φ $s \models \varphi$ iff $\forall \pi \in Paths(s). \pi \models \varphi$	CTL: state formulae existential path quantification $\exists \varphi$ universal path quantification: $\forall \varphi$
complexity of the model checking problems	PSPACE-complete $\mathcal{O}(TS \cdot \exp(\varphi))$	<i>PTIME</i> $\mathcal{O}(TS \cdot \Phi)$
implementation- relation	trace inclusion and the like (proof is PSPACE-complete)	simulation and bisimulation (proof in polynomial time)
fairness	no special techniques needed	special techniques needed

CTL*

CTL*

CTL onde não é obrigatório que um operador LTL $\{X, G, F, U\}$ seja antecedido por um operador A ou E.

Exemplos:

- $A[(pUr) \vee (qUr)],$
- $E(GF\varphi)$
- $A[Xp \vee XXp]$

O CTL* é estritamente mais expressivo que o LTL e o CTL, é computacionalmente muito menos eficiente ...

Sintaxe do CTL*

Fórmulas de Estado

São avaliadas num estado.

$$\varphi ::= \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (A[\alpha]) \mid (E[\alpha])$$

Fórmulas de Caminho

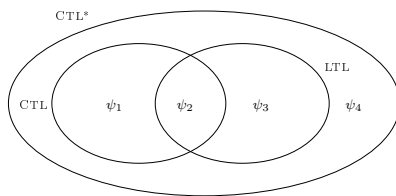
São avaliadas num caminho.

$$\alpha ::= \varphi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha U \alpha) \mid (G\alpha) \mid (F\alpha) \mid (X\alpha)$$

LTL, CTL e CTL*

Uma fórmula α LTL corresponde a $A[\alpha]$ do CTL*. O CTL é o fragmento de CTL* em que

$$\alpha ::= (\alpha U \alpha) \mid (G\alpha) \mid (F\alpha) \mid (X\alpha)$$



$$\begin{aligned} \psi_1 &= \text{AGEF}p \\ \psi_2 &= \text{AG}(p \rightarrow \text{AF}q) \\ \psi_3 &= \text{A}[\text{GF}p \rightarrow \text{F}q] \\ \psi_4 &= \text{E}[\text{GF}p] \end{aligned}$$