La negazione in LTL

Che cosa significa “Non è vero che $\mathsf{F}\,\alpha$”?

La negazione in LTL

Che cosa significa “Non è vero che $\mathsf{F}\,\alpha$”?

Possiamo riformularla così: “Non è vero che in ogni cammino, prima o poi α diventa vera”, cioè “esiste un cammino nel quale α è sempre falsa”.

Che cosa significa $\neg \mathsf{F}\,\alpha$?

La negazione in LTL

Che cosa significa “Non è vero che $\mathrm{F}\,\alpha$”?

Possiamo riformularla così: “Non è vero che in ogni cammino, prima o poi α diventa vera”, cioè “esiste un cammino nel quale α è sempre falsa”.

Che cosa significa $\neg\mathrm{F}\,\alpha$?

“In ogni cammino non è vero che prima o poi α diventa vera”, cioè $\mathrm{G}\,\neg\alpha$

Attenzione: $\neg\mathrm{F}\,\alpha$ **non** è la negazione logica di $\mathrm{F}\,\alpha$.

Limiti espressivi di LTL

LTL non è in grado di esprimere proprietà del tipo

esiste un cammino in cui α

Limiti espressivi di LTL

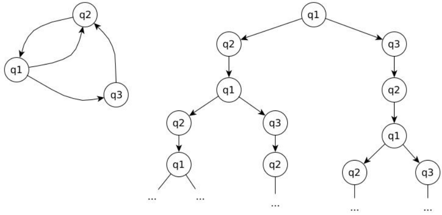
LTL non è in grado di esprimere proprietà del tipo

esiste un cammino in cui α

Esempi:

“è sempre possibile ritornare allo stato iniziale”,
“se il processo P chiede la risorsa, è possibile raggiungere uno stato di *deadlock*”

Alberi di computazione



Computazione: cammino nell'albero a partire dalla radice

Computation Tree Logic – sintassi

Proposizioni atomiche: $AP = \{p_1, p_2, \cdots, q, r, \cdots\}$

Definiamo $\mathrm{FBF}_{\mathrm{CTL}}$

Computation Tree Logic – sintassi

Proposizioni atomiche: $AP = \{p_1, p_2, \cdots, q, r, \cdots\}$

Definiamo $\mathrm{FBF}_{\mathrm{CTL}}$

per ogni $p \in AP$, $p \in \mathrm{FBF}_{\mathrm{CTL}}$

Per ogni $\alpha, \beta \in \mathrm{FBF}_{\mathrm{CTL}}$

1. $\neg\alpha$, $\alpha \vee \beta \in \mathrm{FBF}_{\mathrm{CTL}}$

Computation Tree Logic – sintassi

Proposizioni atomiche: $AP = \{p_1, p_2, \cdots, q, r, \cdots\}$

Definiamo $\mathrm{FBF}_{\mathrm{CTL}}$

per ogni $p \in AP$, $p \in \mathrm{FBF}_{\mathrm{CTL}}$

Per ogni $\alpha, \beta \in \mathrm{FBF}_{\mathrm{CTL}}$

1. $\neg\alpha$, $\alpha \vee \beta \in \mathrm{FBF}_{\mathrm{CTL}}$
2. $AX\,\alpha, EX\,\alpha \in \mathrm{FBF}_{\mathrm{CTL}}$
3. $AF\,\alpha, EF\,\alpha \in \mathrm{FBF}_{\mathrm{CTL}}$
4. $AG\,\alpha, EG\,\alpha \in \mathrm{FBF}_{\mathrm{CTL}}$
5. $A(\alpha\,U\,\beta) \in \mathrm{FBF}_{\mathrm{CTL}}, E(\alpha\,U\,\beta) \in \mathrm{FBF}_{\mathrm{CTL}}$

Computation Tree Logic – sintassi

A e E sono quantificatori sui cammini:

$$\begin{array}{rl} A: & \text{per ogni cammino} \\ E: & \text{esiste un cammino tale che} \end{array}$$

Computation Tree Logic – sintassi

A e E sono quantificatori sui cammini:

$$A: \quad \text{per ogni cammino}$$

$$E: \quad \text{esiste un cammino tale che}$$

“Dopo l’accensione della spia, sarà sempre possibile riportare il sistema allo stato iniziale”

Computation Tree Logic – sintassi

A e E sono quantificatori sui cammini:

A: per ogni cammino

E: esiste un cammino tale che

"Dopo l'accensione della spia, sarà sempre possibile riportare il sistema allo stato iniziale"

s: la spia è accesa
init: il sistema si trova nello stato iniziale

$$AG\,(s \longrightarrow AX\,AG\,\mathrm{EF}\,\mathrm{init})$$

CTL – semantica

Sia $M = (Q, T, I)$ un modello di Kripke, α una formula di CTL, q uno stato di M

$$\boxed{M, q \models \alpha}$$

significa che α è vera nello stato q del modello M

Definiamo la relazione $\models$ per induzione sulla struttura delle formule.

CTL – semantica

Supponiamo che α e β siano due formule, p una proposizione atomica.

CTL – semantica

Supponiamo che α e β siano due formule, p una proposizione atomica.

$M, q \models p \quad$ sse, $\quad p \in I(q)$

$M, q \models \neg\alpha \quad$ sse $\quad M, q \not\models \alpha$

$M, q \models \alpha \vee \beta \quad$ sse $\quad M, q \models \alpha$ o $\pi \models \beta$

CTL – semantica

$M, q \models AX\alpha$ sse per ogni q' tale che $q \to q'$, $M, q' \models \alpha$

CTL – semantica

$M, q \models AX\alpha$ sse per ogni q' tale che $q \to q'$, $M, q' \models \alpha$

$M, q \models EX\alpha$ sse esiste q' tale che $q \to q'$ e $M, q' \models \alpha$

CTL – semantica

$M, q \models AX\alpha$ sse per ogni q' tale che $q \to q'$, $M, q' \models \alpha$

$M, q \models EX\alpha$ sse esiste q' tale che $q \to q'$ e $M, q' \models \alpha$

$M, q \models AU(\alpha, \beta)$ sse per ogni cammino $q = q_0, q_1, \cdots$, esiste $k \geq 0$ tale che $M, q_k \models \beta$, e per ogni i, con $0 \leq i < k$, $M, q_i \models \alpha$

...

Esercizio

Tradurre in LTL i seguenti enunciati

"Chi ruba, presto o tardi finirà in galera"

"Per superare l'esame, è necessario avere studiato"

"Se la cabina è in movimento verso l'alto, si trova all'altezza del secondo piano, ed è stato premuto il pulsante interno di richiesta del quinto piano, la cabina non cambierà direzione fino a quando avrà raggiunto il quinto piano"

Strategia: individuare le proposizioni atomiche; analizzare la struttura della frase, eventualmente riformulandola

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

$$\mathsf{G}\,(hr \longrightarrow \mathsf{XF}\,c)$$

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

$$\mathsf{G}\,(hr \longrightarrow \mathsf{XF}\,c)$$

"Solo chi ruba finirà in galera"

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

$$\mathsf{G}\,(hr \longrightarrow \mathsf{XF}\,c)$$

"Solo chi ruba finirà in galera": $\neg c\,W\,hr$

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

$$\mathsf{G}\,(\mathrm{hr} \longrightarrow \mathsf{X}\mathsf{F}\,\mathrm{c})$$

"Solo chi ruba finirà in galera": $\neg\mathrm{c}\,\mathcal{W}\,\mathrm{hr}$

"Chi ruba finirà in carcere, ma solo dopo avere parlato con un avvocato"

Esercizio

"Chi ruba, presto o tardi finirà in galera"

Proposizioni atomiche: hr = "ho rubato"; c = "sono in carcere"

$$G\,(hr \longrightarrow XF\,c)$$

"Solo chi ruba finirà in galera": $\neg c\,W\,hr$

"Chi ruba finirà in carcere, ma solo dopo avere parlato con un avvocato"

pa = "ho parlato con un avvocato"

$$G\,(hr \longrightarrow (XF\,c \land (\neg c\,U\,pa)))$$

Esercizio

"Se la cabina è in movimento verso l'alto, si trova all'altezza del secondo piano, ed è stato premuto il pulsante interno di richiesta del quinto piano, allora la cabina non cambierà direzione fino a quando avrà raggiunto il quinto piano"

Proposizioni atomiche:

su="cabina sta salendo"
p_i="cabina all'altezza del piano i"
r_i="pulsante interno del piano i è stato premuto"

Esercizio

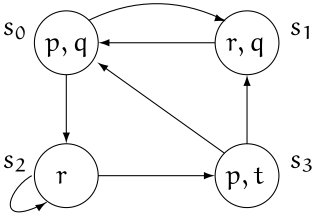
"Se la cabina è in movimento verso l'alto, si trova all'altezza del secondo piano, ed è stato premuto il pulsante interno di richiesta del quinto piano, allora la cabina non cambierà direzione fino a quando avrà raggiunto il quinto piano"

Proposizioni atomiche:

su="cabina sta salendo"
p_i="cabina all'altezza del piano i"
r_i="pulsante interno del piano i è stato premuto"

$$G\left((su \wedge p_2 \wedge r_5) \longrightarrow (su \, U \, p_5)\right)$$

Esercizio



(a) $\mathsf{AF}\, q$

(b) $\mathsf{AG}\,(\mathsf{EF}\,(p \vee q)), \quad \mathsf{GF}\,(p \vee q)$

(c) $\mathsf{EX}\,(\mathsf{EX}\, r), \quad \mathsf{XX}\, r$

(d) $\mathsf{AG}\,(\mathsf{AF}\, q)$

Confronto fra LTL e CTL

Molte proprietà interessanti si possono esprimere sia in LTL sia in CTL

Invarianti

$$\mathsf{AG}\,\neg p \qquad\qquad \mathsf{G}\,\neg p$$

Reattività

$$\mathsf{AG}\,(p \longrightarrow \mathsf{AF}\,q) \qquad\qquad \mathsf{G}\,(p \longrightarrow \mathsf{F}\,q)$$

Confronto fra LTL e CTL

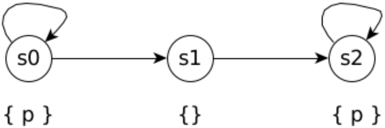
$A G \, E F \, p$ (*reset property*): "da ogni stato raggiungibile in ogni cammino è sempre possibile raggiungere uno stato nel quale vale p"

Questa proprietà non si può esprimere in LTL

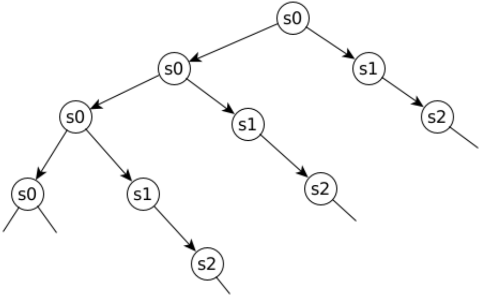
Confronto fra LTL e CTL

FG p: "in ogni cammino, prima o poi si raggiungerà uno stato a partire dal quale p rimane sempre vera"

Questa proprietà non si può esprimere in CTL



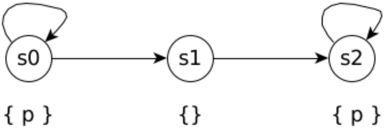
Confronto fra LTL e CTL: FG p



Confronto fra LTL e CTL: $A F A G\, p$

$F G\, p$: "in ogni cammino, prima o poi si raggiungerà uno stato a partire dal quale p rimane sempre vera"

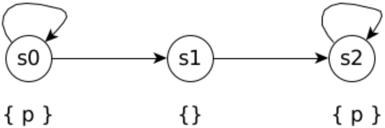
Questa proprietà non si può esprimere in CTL



Confronto fra LTL e CTL: $AF\,AG\,p$

$FG\,p$: “in ogni cammino, prima o poi si raggiungerà uno stato a partire dal quale p rimane sempre vera”

Questa proprietà non si può esprimere in CTL

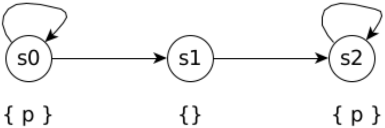


$M, s_0 \models FG\,p \quad M, s_0 \not\models AF\,AG\,p$

Confronto fra LTL e CTL

$\mathsf{FG}\, p$: "in ogni cammino, prima o poi si raggiungerà uno stato a partire dal quale p rimane sempre vera"

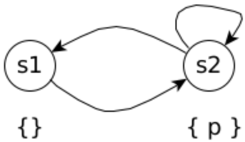
Questa proprietà non si può esprimere in CTL



$M, s_0 \models \mathsf{FG}\, p \quad M, s_0 \not\models \mathsf{AF\,AG}\, p$

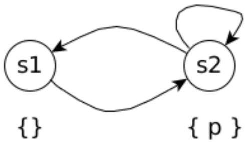
$M, s_0 \models \mathsf{AF\,EG}\, p \quad !$

Confronto fra LTL e CTL



$M, s1 \models AF\,EG\,p$? $M, s1 \models FG\,p$?

Confronto fra LTL e CTL



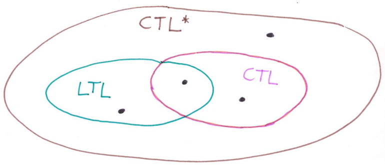
$M, s1 \models \mathrm{AF\,EG}\,p \qquad M, s1 \not\models \mathrm{FG}\,p$

CTL*

La logica CTL* estende sia LTL sia CTL, mantenendo i due quantificatori sui cammini, ma eliminando il vincolo di CTL

$$\mathsf{AFG}\, p \vee \mathsf{AF}\,\mathsf{EG}\, q \qquad \mathsf{EFG}\, q$$

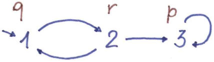
LTL, CTL, CTL*



Equivalenza di modelli rispetto a una logica

Due modelli di Kripke, M_1 e M_2, con stati iniziali q_0 e s_0 si dicono equivalenti rispetto a una logica L se, per ogni formula $\alpha \in FBF_L$

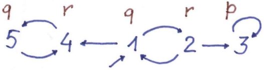
$$M_1, q_0 \models \alpha \qquad \Longleftrightarrow \qquad M_2, s_0 \models \alpha$$



Cammini massimali

$(1\ 2)^\omega$ $\quad$ $(\{q\}\{r\})^\omega$

$(1\ 2)^k 3^\omega$ $\quad$ $(\{q\}\{r\})^k\{p\}^\omega$



$1\ (4\ 5)^\omega$ $\quad$ $\{q\}(\{r\}\{q\})^\omega = (\{q\}\{r\})^\omega$

$(1\ 2)^\omega$ $\quad$ $(\{q\}\{r\})^\omega$

$(1\ 2)^k 3^\omega$ $\quad$ $(\{q\}\{r\})^k\{p\}^\omega$

$(1\ 2)^k 1 (4\ 5)^\omega$ $\quad$ $(\{q\}\{r\})^k(\{q\})(\{r\}\{q\})^\omega =$

$= (\{q\}\{r\})^\omega$