Ragionamento Automatico
Logiche Temporali: LTL

Lezione 10

Sommario

♢ Capitolo 3 del libro di M. Huth e M. Ryan: *Logic in Computer Science: Modelling and reasoning about systems* (Second Edition) Cambridge University Press, 2004.

♢ Verifica/certificazione di sistemi dinamici

♢ Logiche Temporali

♢ Model Checking

Motivazioni

♢ Sistemi **Safety Critical**

♢ Sistemi **Commercially Critical**

♢ Sistemi **Mission Critical**

Tecniche di verifica formale

♢ Basate su dimostrazione vs basate su modello
($\Gamma \vdash \phi$ vs $\mathcal{M} \models \phi$)
♢ Grado di automazione:
automatico vs interattivo
♢ Verifica del Comportamento complessivo vs Proprietà
♢ Dominio: Hardware/Software;
sistema che termina/sistema reattivo
(sistema operativo, sistemi embedded, robot...)
♢ Verifica Pre vs Post Sviluppo

Model Checking

Il **model checking** è un metodo **automatico, basato su modello, per fare verifica di proprietà**

È inteso per la verifica di **sistemi concorrenti reattivi**.

Nasce come metodo di verifica post sviluppo

Logica Temporale LTL

♢ **LTL– Linear Time Logic**
Sia p un atomo proposizionale elemento di un qualche insieme $\mathcal{A}$ di Atomi. Una formula di LTL è definita in BNF come segue

$$\phi ::= \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\mathbf{X}\phi) \mid (\mathbf{F}\phi) \mid (\mathbf{G}\phi) \mid (\phi \mathbf{U} \phi) \mid (\phi \mathbf{W} \phi) \mid (\phi \mathbf{R} \phi)$$

X sta per "ne**X**t"
F sta per "exists **F**uture"
G sta per "all future"
U sta per "**U**ntil "
R sta per "**R**elease "
W sta per "**W**eak Until"

Precedenza degli operatori

1) $\neg, \mathbf{X}, \mathbf{F}, \mathbf{G}$
2) $\mathbf{U}, \mathbf{R}, \mathbf{W}$
3) $\wedge, \vee$
4) $\rightarrow$

Esempi

1) $\mathbf{F}p \wedge \mathbf{G}q \rightarrow p\mathbf{W}r$
2) $\mathbf{F}(p \rightarrow \mathbf{G}r) \vee \neg q \mathbf{U} p$
3) $p\mathbf{W}(q\mathbf{W}r)$
4) $\mathbf{GF}p \rightarrow \mathbf{F}(q \vee s)$

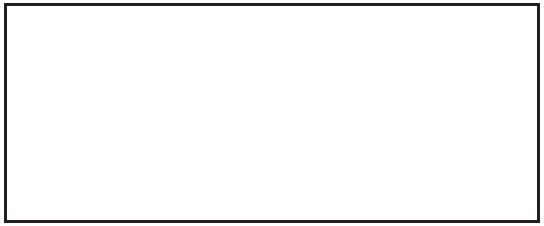
Semantica di LTL

LTL viene modellata mediante sistemi di transizioni (automi a stati finiti)

Sia $\mathcal{S}$ un **insieme di stati**
Sia $\rightarrow$ una **relazione binaria** tra stati, i.e. $\rightarrow \subseteq \mathcal{S} \times \mathcal{S}$ t.c. per ogni $s \in \mathcal{S}$ esiste un $s' \in \mathcal{S}$ per cui vale $s \rightarrow s'$
Sia $L : \mathcal{S} \rightarrow 2^A$ una **funzione di etichettatura** che associa stati a sottoinsiemi di atomi.
Un **sistema di transizioni** è una terna

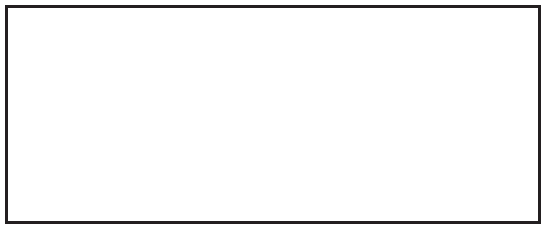
$$\mathcal{M} = \langle \mathcal{S}, \rightarrow, L \rangle$$

Esempio



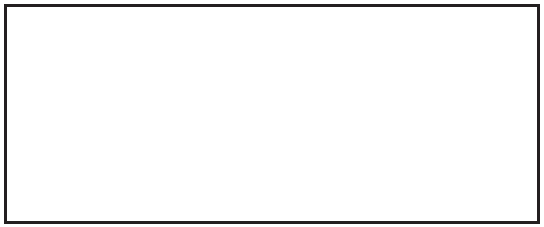
dove $\mathcal{S} = \{s\_0, s\_1, s\_2\}$,
$A = \{p, q, r\}$, $L(s\_0) = \{p, q\}$, $L(s\_1) = \{q, r\}$, $L(s\_2) = \{r\}$,
$\rightarrow = \{\langle s\_0, s\_1\rangle, \langle s\_0, s\_2\rangle, \langle s\_1, s\_0\rangle, \langle s\_1, s\_2\rangle, \langle s\_2, s\_2\rangle, \}$

Esempio



Se c'è uno stato di deadlock

Esempio



aggiungo un nodo con un ciclo

Traccia

Una **traccia (path)** in un modello $\mathcal{M} = \langle \mathcal{S}, \rightarrow, L \rangle$ è una sequenza **infinita** di stati

$$s\_1, s\_2, s\_3, \cdots$$

tali che per ogni $i \geq 1, s\_i \rightarrow s\_(i+1)$. La scriviamo

$$\pi = s\_1 \rightarrow s\_2 \rightarrow s\_3, \cdots$$

Una traccia rappresenta un **possibile futuro** del sistema.
Con la notazione π^i indichiamo il **suffisso** di π che inizia con l'elemento i-esimo, e.g.:

$$\pi^3 = s\_3 \rightarrow s\_4 \rightarrow s\_5, \cdots$$

Soddisfacibilità di formule LTL

Sia $\mathcal{M} = \langle \mathcal{S}, \rightarrow, L \rangle$ un modello e $\pi = s\_1 \rightarrow s\_2, \cdots$ una traccia in $\mathcal{M}$. La traccia π soddisfa una formula LTL p (e scriviamo $\pi \models p$)sse:

1. $\pi \models \top$
2. $\pi \not\models \bot$
3. $\pi \models p$ sse $p \in L(s\_1)$
4. $\pi \models \neg\phi$ sse $\pi \not\models \phi$
5. $\pi \models \phi_1 \wedge \phi_2$ sse $\pi \models \phi_1$ e $\pi \models \phi_2$
6. $\pi \models \phi_1 \vee \phi_2$ sse $\pi \models \phi_1$ oppure $\pi \models \phi_2$
7. $\pi \models \phi_1 \rightarrow \phi_2$ sse $\pi \models \phi_2$ qualora $\pi \models \phi_1$
8. $\pi \models \mathbf{X}\phi$ sse $\pi^2 \models \phi$
9. $\pi \models \mathbf{G}\phi$ sse per ogni $i \geq 1$ $\pi^i \models \phi$
10. $\pi \models \mathbf{F}\phi$ sse esiste un $i \geq 1$ t.c. $\pi^i \models \phi$

continua

Soddisfacibilità di formule LTL (cont)

11. $\pi \models \phi\mathbf{U}\psi$ sse esiste un $i \geq 1$ t.c. $\pi^i \models \psi$
    e per ogni $j = 1, \cdots, i-1$ si ha $\pi^j \models \phi$
12. $\pi \models \phi\mathbf{W}\psi$ sse o esiste un $i \geq 1$ t.c. $\pi^i \models \psi$
    e per ogni $j = 1, \cdots, i-1$ si ha $\pi^j \models \phi$;
    oppure per tutti i $k \geq 1$ si ha $\pi^k \models \phi$
13. $\pi \models \phi\mathbf{R}\psi$ sse o esiste un $i \geq 1$ t.c. $\pi^i \models \phi$
    e per ogni $j = 1, \cdots, i$ si ha $\pi^j \models \psi$;
    oppure per tutti i $k \geq 1$ si ha $\pi^k \models \psi$

Commenti

♢ 3) Un atomo viene valutato nel primo stato della traccia
♢ 8) Con l'operatore next si leva il primo elemento della traccia e si usa π^2
♢ In generale, in una formula con operatori temporali si guarda al primo stato della traccia o di suoi opportuni suffissi
♢ 11) Notare come viene definito l'until: esiste uno stato nella traccia in cui ψ diventa vero, e prima del quale ϕ è sempre vero. Magari ϕ è vero anche dopo, magari ψ era vero qualche volta anche prima
♢ 12) Questa ci dice che finché non viene a valere ψ vale ϕ, ma ψ potrebbe anche non valere mai
♢ **R** è molto simile a **W**
♢ **R** è duale di **U**, i.e.: $\phi\mathbf{R}\psi \equiv \neg(\neg\phi\mathbf{U}\neg\psi)$
♢ con questa definizione, il futuro include il presente

Definizione

Sia $\mathcal{M} = \langle \mathcal{S}, \rightarrow, L \rangle$ un modello, $s \in \mathcal{S}$ uno stato e ϕ una formula LTL.
Scriviamo $\mathcal{M}, s \models \phi$ (o, più brevemente $s \models \phi$) se per ogni traccia π che origina in s abbiamo $\pi \models \phi$.

Esempi

$\mathcal{M}, s_0 \models p \wedge q$ vale
$\pi \models p \wedge q$ vale per ogni traccia π che inizia con s_0
$\mathcal{M}, s_0 \models r$ non vale
$\mathcal{M}, s_0 \models \neg r$ vale
$\mathcal{M}, s_0 \models \top$ vale, per definizione
$\mathcal{M}, s_0 \models \mathbf{X} r$ vale perché r è in s_1 e in s_2
$\mathcal{M}, s_0 \models \mathbf{X} q$ non vale perché q è in s_1 ma non in s_2

Esempi

$\mathcal{M}, s_0 \models \mathbf{X}(q \wedge r)$ non vale

$\mathcal{M}, s_0 \models \mathbf{G}\neg(p \wedge r)$ vale perché $\neg(p \wedge r)$ vale in
tutti gli stati raggiungibili a partire da s_0

Esempi

$\mathcal{M}, s_0 \models \mathbf{X} r$ vale
$\mathcal{M}, s_0 \models \mathbf{G} r$ non vale
$\mathcal{M}, s_2 \models \mathbf{X} r$ vale
$\mathcal{M}, s_2 \models \mathbf{G} r$ vale perché r vale in tutti gli stati raggiungibili a partire da s_2

Esempi

FGϕ vuol dire che esiste un momento futuro dopo il quale ϕ sarà sempre vera. Trovare un esempio sull'automa in figura

GFϕ vuol dire che per tutto il futuro, ogni tanto ϕ tornerà a essere vera. Trovare un esempio sull'automa in figura

Esempi più pratici 1

G$\neg(started \wedge \neg ready))$
È impossibile raggiungere uno stato in cui $started$ vale e $ready$ non vale.

G$(requested \rightarrow$ **F**$acknowledged)$
Se in uno stato arriva una richiesta, prima o poi verrà soddisfatta.

GF$enabled$
Un processo viene abilitato ($enabled$) infinite volte in ogni traccia di computazione

Esempi più pratici 2

FG$deadlock$
Da un certo punto in poi un processo sarà permanentemente in deadlock

GF$enabled \rightarrow$ **GF**$running$
Se un proscesso è abilitato infinite volte, allora gira infinite volte

G$(floor2 \wedge directionup \wedge buttonpressed5 \rightarrow (directionup$ **U** $floor5))$
Se l'ascensore sta al secondo piano e la direzione è up, e lo chiamano dal quinto piano, non cambia direzione fin quando non è arrivato al quinto piano

Equivalenze rilevanti 1

Diciamo che due formule LTL ϕ e ψ sono **logicamente equivalenti**, e scriviamo $\phi \equiv \psi$ sse per ogni $\mathcal{M}$ e per ogni π in $\mathcal{M}$ si ha che

$$\pi \models \phi \text{ sse } \pi \models \psi$$

Esempi notevoli:
$\neg$**G**$\phi \equiv$ **F**$\neg\phi$
$\neg$**F**$\phi \equiv$ **G**$\neg\phi$
G e **F** sono duali

$\neg$**X**$\phi \equiv$ **X**$\neg\phi$
X è duale di sé stesso

Equivalenze rilevanti 2

$\neg(\phi \mathbf{U} \psi) \equiv \neg\phi \mathbf{R} \neg\psi$
$\neg(\phi \mathbf{R} \psi) \equiv \neg\phi \mathbf{U} \neg\psi$
U e **R** sono duali

$\mathbf{F}(\phi \vee \psi) \equiv \mathbf{F}\phi \vee \mathbf{F}\psi$
$\mathbf{G}(\phi \wedge \psi) \equiv \mathbf{G}\phi \wedge \mathbf{G}\psi$
F distribuisce rispetto a $\vee$;
G distribuisce rispetto a $\wedge$

Nota: **F** non distribuisce rispetto a $\wedge$
(esempio $\mathbf{F}p \wedge \mathbf{F}r$ non implica $\mathbf{F}(p \wedge r)$).

Equivalenze rilevanti 3

$\mathbf{F}\phi \equiv \top \mathbf{U} \phi$
Prima o poi ϕ diventa vero
$\mathbf{G}\phi \equiv \bot \mathbf{R} \phi$
ϕ non viene mai rilasciato, quindi rimane sempre vero

$\phi \mathbf{U} \psi \equiv \phi \mathbf{W} \psi \wedge \mathbf{F}\psi$
L'until "forte" è come l'until "debole", più la condizione che prima o poi succederà
$\phi \mathbf{W} \psi \equiv \phi \mathbf{U} \psi \vee \mathbf{G}\psi$

$\phi \mathbf{W} \psi \equiv \psi \mathbf{R} (\phi \vee \psi)$
$\phi \mathbf{R} \psi \equiv \psi \mathbf{W} (\phi \wedge \psi)$