Ragionamento Automatico
Logiche temporali: CTL

Lezione 11

Sommario

♢ Capitolo 3 paragrafi 4 e 5 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

Logica Temporale CTL

Motivazioni: mancanza di potere espressivo di LTL

♢ In LTL non si può quantificare sulle tracce

Non si può dire "esiste una traccia in cui vale ϕ"
Si potrebbe dire "Per ogni traccia vale $\neg\phi$"
Ma se voglio alternare i quantificatori non posso.

La sintassi di CTL

Sia p un atomo proposizionale elemento di un qualche insieme $\mathcal{A}$ di Atomi. Una formula di CTL è definita in BNF come segue

$$\begin{aligned}\phi \ ::= \ & \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \\ & \mid \mathbf{AX}\phi \mid \mathbf{EX}\phi \mid \mathbf{AF}\phi \mid \mathbf{EF}\phi \\ & \mid \mathbf{AG}\phi \mid \mathbf{EG}\phi \mid \mathbf{A}[\phi\mathbf{U}\phi] \mid \mathbf{E}[\phi\mathbf{U}\phi]\end{aligned}$$

X sta per "ne**X**t"
F sta per "exists **F**uture"
G sta per "all future"
U sta per "**U**ntil "
A sta per "**A**ll paths"; notare le parentesi quadre
E sta per "**E**xists a path"; notare le parentesi quadre

Nota

Gli operatori modali di CTL vanno sempre a coppia: non ha senso scrivere **A** o **E** da soli, come non ha senso scrivere **X**, **F** ,**G** ,**U** da soli.

Gli operatori **AX**, **EX**, **AF**, **EF**, **AG**, **EG** sono unari
Gli operatori **AU**, **EU** sono binari, e si usano con una notazione mista: in parte prefissa e in parte infissa

Gli operatori **W** e **R** si possono derivare

Precedenza degli operatori

1) $\neg$, **AX**, **EX**, **AF**, **EF**, **AG**, **EG**,
2) $\wedge, \vee$
3) $\rightarrow$, **AU**, **EU**,

Esempi

1) **EFE**$[r\mathbf{U}q]$
2) **A**$[p\mathbf{U}$ **EF**$r]$
3) **EFEG**$p \rightarrow$ **AF**r che lega come (**EFEG**$p) \rightarrow$ **AF**r
   e non come **EF**(**EG**$p \rightarrow$ **AF**$r)$
   e neppure come **EFEG**$(p \rightarrow$ **AF**$r)$
4) **A**$[p_1\mathbf{U}$ **A**$[p_2\mathbf{U}p_3]]$
5) **E**$[$**A**$[p_1\mathbf{U}p_2]\mathbf{U}p_3]$

Notare che invece **A**$[(r\mathbf{U}q) \wedge (p\mathbf{U}r)]$ non è una formula ben formata, mentre **A**$[(r \wedge q)\mathbf{U}(p \rightarrow r)]$ lo è.

Soddisfacibilità di formule CTL

Sia $\mathcal{M} = \langle \mathcal{S}, \rightarrow, L \rangle$ un modello per CTL, sia $s \in \mathcal{S}$ e sia ϕ una formula CTL. Diciamo che $\mathcal{M}, s \models \phi$ sse:

1. $\mathcal{M}, s \models \top$
2. $\mathcal{M}, s \not\models \bot$
3. $\mathcal{M}, s \models p$ sse $p \in L(s)$
4. $\mathcal{M}, s \models \neg\phi$ sse $s \not\models \phi$
5. $\mathcal{M}, s \models \phi_1 \wedge \phi_2$ sse $s \models \phi_1$ e $s \models \phi_2$
6. $\mathcal{M}, s \models \phi_1 \vee \phi_2$ sse $s \models \phi_1$ oppure $s \models \phi_2$
7. $\mathcal{M}, s \models \phi_1 \rightarrow \phi_2$ sse $s \models \phi_2$ qualora $s \models \phi_1$
8. $\mathcal{M}, s \models$ **AX**ϕ sse per ogni s_1 t.c. $s \rightarrow s_1$, si ha $\mathcal{M}, s_1 \models \phi$
   (cioè **AX** dice: "in ogni stato successivo")
9. $\mathcal{M}, s \models$ **EX**ϕ sse esiste un s_1 t.c. $s \rightarrow s_1$, tale che $\mathcal{M}, s_1 \models \phi$
   (cioè **EX** dice: "in qualche stato successivo")

continua

Soddisfacibilità di formule CTL (cont)

10. $\mathcal{M}, s \models \mathbf{AG}\phi$ sse per ogni traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    e per ogni s_i nella traccia, si ha $\mathcal{M}, s_i \models \phi$
    (cioè, per ogni computazione che parte da s,
    ϕ vale globalmente)
11. $\mathcal{M}, s \models \mathbf{EG}\phi$ sse esiste una traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    e per ogni s_i nella traccia, si ha $\mathcal{M}, s_i \models \phi$
    (cioè, esiste una computazione che parte da s
    in cui ϕ vale globalmente)

continua

Soddisfacibilità di formule CTL (cont)

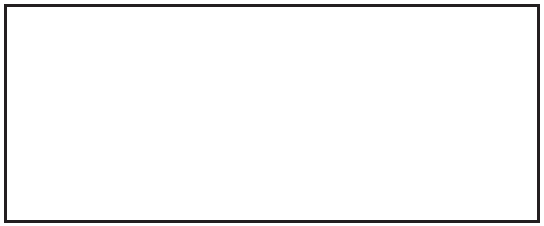
12. $\mathcal{M}, s \models \mathbf{AF}\phi$ sse per ogni traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    esiste un s_i nella traccia tale che $\mathcal{M}, s_i \models \phi$
    (cioè, ogni computazione che parte da s,
    contiene uno stato in cui vale la formula ϕ)
13. $\mathcal{M}, s \models \mathbf{EF}\phi$ sse esiste una traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    ed esiste un s_i nella traccia tale che $\mathcal{M}, s_i \models \phi$
    (cioè, esiste una computazione che parte da s,
    che contiene uno stato in cui vale la formula ϕ)

continua

Soddisfacibilità di formule CTL (fine!)

14. $\mathcal{M}, s \models \mathbf{A}[\phi \mathbf{U} \psi]$ sse ogni traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    soddisfa $[\phi \mathbf{U} \psi]$, cioè, sse esiste un $i \geq 1$
    t.c. $s^i \models \psi$ e per ogni $j = 1, \cdots, i-1$ si ha $s^j \models \phi$
    (cioè, ogni computazione che parte da s
    soddisfa ϕ fin quando non vale ψ)
15. $\mathcal{M}, s \models \mathbf{E}[\phi \mathbf{U} \psi]$ sse esiste una traccia $s = s_1 \rightarrow s_2 \rightarrow s_3 \ldots$
    che soddisfa $[\phi \mathbf{U} \psi]$, cioè, sse esiste un $i \geq 1$
    t.c. $s^i \models \psi$ e per ogni $j = 1, \cdots, i-1$ si ha $s^j \models \phi$
    (cioè, esiste una computazione che parte da s
    che soddisfa ϕ fin quando non vale ψ)

Esempio 1



Sistema il cui stato iniziale soddisfa $\mathbf{AF}fbf$

Esempio 2

Sistema il cui stato iniziale soddisfa $\mathbf{EF}fbf$

Esempio

Sistema il cui stato iniziale soddisfa $\mathbf{AG}fbf$

Esempio

Sistema il cui stato iniziale soddisfa $\mathbf{EG}fbf$

Riprendiamo il primo esempio di sistema

$\mathcal{M}, s_0 \models p \wedge q$ vale
$\mathcal{M}, s_0 \models \neg r$ vale
$\mathcal{M}, s_0 \models \top$ vale
$\mathcal{M}, s_0 \models \mathbf{EX}(q \wedge r)$ vale
$\mathcal{M}, s_0 \models \neg\mathbf{AX}(q \wedge r)$ vale
$\mathcal{M}, s_0 \models \mathbf{EF}(q \wedge r)$ vale
$\mathcal{M}, s_0 \models \neg\mathbf{EF}(p \wedge r)$ vale

Esempi (cont)

$\mathcal{M}, s_2 \models \mathbf{EG}r$ vale
$\mathcal{M}, s_2 \models \mathbf{AG}r$ vale
$\mathcal{M}, s_0 \models \mathbf{AF}r$ vale
$\mathcal{M}, s_0 \models \mathbf{E}[(p \wedge q)\mathbf{U}r]$ vale
$\mathcal{M}, s_0 \models \mathbf{A}[p\mathbf{U}r]$ vale
$\mathcal{M}, s_0 \models \mathbf{A}[q\mathbf{U}r]$ vale
$\mathcal{M}, s_0 \models \mathbf{AG}(p \wedge q \wedge r \rightarrow \mathbf{EFEG}r)$ vale
$\mathcal{M}, s_0 \models \mathbf{AG}(p \vee q \vee r \rightarrow \mathbf{EFEG}r)$ vale

Equivalenze rilevanti

Diciamo che due formule CTL 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:
$\neg\mathbf{AF}\phi \equiv \mathbf{EG}\neg\phi$
$\neg\mathbf{EG}\phi \equiv \mathbf{AF}\neg\phi$
$\neg\mathbf{AX}\phi \equiv \mathbf{EX}\neg\phi$
e inoltre:
$\mathbf{AF}\phi \equiv \mathbf{A}[\top\mathbf{U}\phi]$
$\mathbf{EF}\phi \equiv \mathbf{E}[\top\mathbf{U}\phi]$

Equivalenze rilevanti, segue

$$\begin{array}{rcl}
\mathbf{AG}\phi & \equiv & \phi \wedge \mathbf{AXAG}\phi \\
\mathbf{EG}\phi & \equiv & \phi \wedge \mathbf{EXEG}\phi \\
\mathbf{AF}\phi & \equiv & \phi \vee \mathbf{AXAF}\phi \\
\mathbf{EF}\phi & \equiv & \phi \vee \mathbf{EXEF}\phi \\
\mathbf{A}[\phi\mathbf{U}\psi] & \equiv & \psi \vee (\phi \wedge \mathbf{AXA}[\phi\mathbf{U}\psi]) \\
\mathbf{E}[\phi\mathbf{U}\psi] & \equiv & \psi \vee (\phi \wedge \mathbf{EXE}[\phi\mathbf{U}\psi])
\end{array}$$

La Logica Temporale CTL*

Il potere espressivo di LTL e CTL:

♢ In LTL non si può quantificare sulle tracce, in CTL si.

♢ In CTL non posso dire $\mathbf{F}p \rightarrow \mathbf{F}q$, che invece dico in LTL.

♢ CTL* combina il potere espressivo di entrambe, rimuovendo il vincolo che gli operatori di LTL devono essere sempre abbinati ad **A** ed **E**.

La Logica Temporale CTL*

Per cui, per esempio si può dire.

$\mathbf{A}[(p\mathbf{U}r) \vee (q\mathbf{U}r)]$
$\mathbf{A}[\mathbf{X}p \vee \mathbf{XX}p]$
$\mathbf{E}[\mathbf{GF}p]$

La prima si può in effetti esprimere in modo molto complesso in LTL, la seconda e la terza no.

La sintassi di CTL*

La definizione delle formule in CTL* è **mutuamente ricorsiva** e si basa sulla nozione di **formule di stato** e **formule di traccia**. Le prime sono valutate negli stati, le seconde lungo le tracce.

$$\phi \ ::= \ \top \mid p \mid \neg(\phi) \mid (\phi \wedge \phi) \mid \mathbf{A}[\alpha] \mid \mathbf{E}[\alpha]$$

dove p è una formula atomica e α è una formula di traccia.

$$\alpha \ ::= \ \phi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha\mathbf{U}\alpha) \mid (\mathbf{G}\alpha) \mid (\mathbf{F}\alpha) \mid (\mathbf{X}\alpha)$$

dove ϕ è una formula di stato.

Il potere espressivo

CTL* è piu' espressiva di LTL e CTL, ma più complessa da maneggiare