Verifica Automatica

Domande di Teoria

1 Dare sintassi e semantica delle formule CTL

Sintassi.

$$\Phi := true \mid a \mid \Phi_1 \ \wedge \Phi_2 \mid \neg\Phi \mid \exists\Psi \mid \forall\Psi$$
$$\Psi := \circ\Phi \mid \Phi_1 \,\mathcal{U}\, \Phi_2 \mid \ \diamond\, \Phi \mid \Box\Phi$$

Semantica. Dato $M = \langle S, N, V\rangle$, $M \models S \times WFF$ è definita come:

1. $M, s \not\models \bot$
2. $M, s \models p$ iff $p \in V(s)$
3. $M, s \models A \wedge B \iff M, s \models A \wedge M, s \models B$
4. $M, s \models A \vee B \iff M, s \models A \vee M, s \models B$
5. $M, s \models \neg A \iff M, s \not\models A$
6. $M, s \models A \to B \iff (M, s \models A \to M, s \models B)$
7. $M, s \models \forall\Box A \iff \forall b_s \forall s' \in b_s \ M, s' \models A$
8. $M, s \models \forall \diamond A \iff \forall b_s \exists s' \in b_s \ M, s' \models A$
9. $M, s \models \exists\Box A \iff \exists b_s \forall s' \in b_s \ M, s' \models A$
10. $M, s \models \exists \diamond A \iff \exists b_s \exists s' \in b_s \ M, s' \models A$
11. $M, s \models \forall \bigcirc A \iff \forall s'(sNs' \to M, s' \models A)$
12. $M, s \models \exists \bigcirc A \iff \exists s'(sNs' \wedge M, s' \models A)$
13. $M, s \models B \,\exists\mathcal{U} A \iff \exists b_s, \exists k(M, b_s[k] \models A \wedge \forall j \in [0, k-1] b_s[j] \models B)$
14. $M, s \models B \,\forall\mathcal{U} A \iff \forall b_s, \exists k(M, b_s[k] \models A \wedge \forall j \in [0, k-1] b_s[j] \models B)$

2 Si definiscano gli automi di Buchi generalizzati. Dato un automa generalizzato di Buchi B, si definisca il linguaggio accettato da B.

Un automa di Buchi non deterministico generalizzato è una tupla

$$G = \langle Q, \Sigma, \delta, Q_0, \mathcal{F} \rangle$$

dove

- Q è l'insieme degli stati (finito);
- Σ è l'alfabeto di simboli utilizzati;
- $\delta : Q \times \Sigma \to 2^Q$
- $Q_0 \subseteq Q$ è l'insieme degli stati inziali;
- $\mathcal{F} \subseteq 2^Q$ è l'insieme degli accept set.

Una run $q_0, q_1, \ldots, q_n$ per una parola $A_0 A_1 \cdots \in \Sigma^\omega$ è un path $\pi = q_0 q_1 \ldots$ dove $q_0 \in Q_0$ e $q_{i+1} \in \delta(q_i, A_i)$.

Una run si dice accettante se ogni accept set è visitato infinitamente spesso, ovvero se

$$\forall F \in \mathcal{F} \overset{\infty}{\exists} i \in \mathbb{N} \; s.t. q_i \in F$$

Il linguaggio generato da tali automi è

$$L_\omega(G) = \{\sigma \in \Sigma^\omega : \sigma \text{ ha una run accettante in } G\}$$

Notare che, se non ci sono stati di accettazione, un GNBA accetta tutte le possibili parole, mentre un NBA (che al posto di $\mathcal{F}$ ha solo un insieme F di stati finali) non accetta nulla.

3 Dare sintassi e semantica delle formule LTL. Mostrare come l'operatore $\Box$ sia definito a partire dall'until.

Sintassi. La sintassi delle formule LTL è così composta:

$$\phi ::= \; a \mid true \mid \phi_1 \vee \phi_2 | \neg \, \phi \mid \; \bigcirc \; \phi | \; \phi_1 \mathcal{U} \phi_2$$

dove $a \in AP$.

Semantica. Viene data la semantica per $\sigma = A_0 A_1 \cdots \in (2^{AP})^\omega$.

- $\sigma \models true$
- $\sigma \models a$ if $A_0 \models a$ $(a \in A_0)$
- $\sigma \models \phi_1 \vee \phi_2$ if $\sigma \models \phi_1$ or $\sigma \models \phi_2$
- $\sigma \models \neg\phi$ if $\sigma \not\models \phi$
- $\sigma \models \bigcirc\phi$ if suffix$(\sigma, 1) = A_1 A_2 A_3 \ldots \models \phi$
- $\sigma \models \phi_1 \mathcal{U} \phi_2$ se esiste $j \geq 0$ tale che:
  - suffix$(\sigma, j) = A_j A_{j+1} A_{j+2} \ldots \models \phi_2 \wedge$
  - suffix$(\sigma, i) = A_i A_{i+1} A_{i+2} \ldots \models \phi_1$ for $0 \geq i \geq j$
- $\sigma \models \diamond\phi$ se e solo se $\exists j \geq 0$ tale che $A_j A_{j+1} A_{j+2} \ldots \models \phi$
- $\sigma \models \Box\phi$ se e solo se $\forall j \geq 0$ tale che $A_j A_{j+1} A_{j+2} \ldots \models \phi$

L'operatore $\Box$ è definito dal weak-until come $\phi \, W \, false$

4 Si diano le definizioni di unconditional LTL-fairness, weak LTL-fairness e strong LTL-fairness. Cosa è una traccia fair per un TS?

- ρ is uncond. fair se $\overset{\infty}{\exists} i \geq 0 . \alpha_i \in A$;
- ρ is strongly fair se $\overset{\infty}{\exists} i \geq 0 . A \cap Act(s_i) \neq \emptyset \implies \overset{\infty}{\exists} i \geq 0 . \alpha_i \in A$
- ρ is strongly fair se $\overset{\infty}{\forall} i \geq 0 . A \cap Act(s_i) \neq \emptyset \implies \overset{\infty}{\exists} i \geq 0 . \alpha_i \in A$

Una traccia ρ è $\mathcal{F} - fair$ se, data una fairness assumption

$$\mathcal{F} = \langle \mathcal{F}_{strong}, \mathcal{F}_{weak}, \mathcal{F}_{ucond} \rangle$$

vale che:

- ρ è uncond. fair;
- ρ è strongly fair;
- ρ è weakly fair.

per tutte le $A \in F - \ldots$.

5 Si dia la definizione di TS. Si definiscano i concetti di cammino infinito e di proprietà di un TS.

Un transition system è una tupla

$$T = (S, Act, \rightarrow, S_0, AP, L)$$

dove

- S è un insieme di stati;
- Act è l'insieme delle azioni;
- $\rightarrow \subseteq S \times Act \times S$ è la relazione di transizione;
- S_0 è lo stato iniziale;
- AP è l'insieme delle *atomic propositions*
- $L : S \rightarrow 2^{AP}$ è la funzione di labeling.

Un **cammino infinito** è una sequenza di stati $\pi = s_0 s_1 \dots$ di lunghezza infinita.

6 Dare sintassi e semantica delle formule LTL rispetto ai TS.

Considero solo tracce infinite. Dato un TS senza stati terminali, una formula su $AP\phi$, l'interpretazione di ϕ su cammini infiniti è

$$\pi = s_0 s_1 \dots \models \phi \iff trace(\pi) \models \phi \iff trace(\pi) \in Words(\phi)$$

dove

$$Words(\phi) = \{\sigma \in (2^{AP})^{\omega} : \sigma \models \phi\}$$