

Bounded Linear Temporal Logic

Giorgio Mariani

Sommario

In questo documento vengono presentate sintassi e semantica della *Bounded Linear Temporal Logic* (*BLTL*) e della *BLTL**, logiche derivate della nota *Linear Temporal Logic*.

1 Introduzione

Una formula *BLTL* (oppure una formula *BLTL**) è utilizzata per descrivere una certa proprietà "dilatata" nel tempo. La validità della formula a differenza di altre logiche è definita rispetto ad un istante e dipende dagli istanti temporali successivi a questo.

2 Definizioni

Prima di poter descrivere sintassi e semantica di una formula *BLTL* occorre dare delle definizioni preliminari.

Insieme delle variabili Definiamo \mathcal{V} come l'insieme delle stringhe di lunghezza finita sopra l'alfabeto inglese.

Traccia di simulazione Una traccia di simulazione $\sigma : Var \times [s, h] \rightarrow \mathbb{R}$ (con $Var \subset \mathcal{V}$ e $s \leq h$) è una funzione che definisce il valore di una certa variabile ad un certo istante di tempo.

Inizio e orizzonte di simulazione Definiamo le costanti reali s e h ($s \leq h$) rispettivamente come l'istante iniziale e finale di una traccia di simulazione, ovvero gli estremi (rispettivamente sinistro e destro) del dominio della data traccia. Il valore h è noto anche come l'*orizzonte* di simulazione.

3 Sintassi *BLTL*

Sono di seguito elencati i costrutti che generano il linguaggio *BLTL*:

$\rho(t)$	$f_1 \mathbf{U}^\alpha f_2$
$f_1 \vee f_2$	$\Box^\alpha f_1$
$f_1 \wedge f_2$	$\Diamond^\alpha f_1$
$\neg f_1$	

Con ρ predicato, $\alpha \in \mathbb{R}$ e f_1, f_2 sotto formule *BLTL*.

Predicati nel dettaglio

Più specificatamente ρ è un predicato il cui valore dipende dall'istante di tempo considerato, in particolare ρ deve essere in una delle seguenti forme:

$$\begin{aligned}\rho &= \sum_{i=0}^n \alpha_i x_i \mathcal{R} \beta \\ \rho &= True \\ \rho &= False\end{aligned}$$

Tale per cui:

- α_i è una sequenza di costanti reali.
- β è costante reale.
- x_i è una sequenza di nomi di variabile in \mathcal{V} .
- \mathcal{R} è una relazione binaria che può assumere valore uguale a $<, \leq, >, \geq, = \neq$.

In altre parole ρ può essere una relazione tra una combinazione lineare ed un valore costante, oppure può essere una costante booleana.

Valore di un predicato

Data una traccia di simulazione μ diremo che $\rho(t)$ è vero se:

- Nel caso in cui $\rho = \sum_{i=0}^n \alpha_i x_i \mathcal{R} \beta$, allora questo è valido per un istante t se e solo se:

$$\sum_{i=0}^n \alpha_i \cdot \mu(x_i, t) \mathcal{R} \beta$$

- Nel caso in cui $\rho = True$ allora $\rho(t)$ è valido.
- Nel caso in cui $\rho = False$ allora $\rho(t)$ non è valido.

4 Semantica BLTL

Consideriamo una formula BLTL ϕ , una traccia di simulazione μ ed un istante t , vorremo poter dire se la formula ϕ è vera oppure falsa rispetto a t data la traccia μ (espresso tramite le notazioni $\mu, t \models \phi$ e $\mu, t \not\models \phi$ rispettivamente).

Tempo minimo di simulazione

Per poter definire la validità di una formula occorre prima esprimere la funzione $minTime : BLTL \rightarrow \mathbb{R}$. La funzione prende in input una formula BLTL ϕ e restituisce la minima durata ($h - s$) che una traccia μ deve avere per poterne valutare almeno un istante.

Definizione a casi di $minTime$:

$$minTime(\phi) = \begin{cases} 0 & \text{se } \phi = \rho \\ minTime(f_1) & \text{se } \phi = \neg f_1 \\ max(minTime(f_1), minTime(f_2)) & \text{se } \phi = f_1 \vee f_2 \\ max(minTime(f_1), minTime(f_2)) & \text{se } \phi = f_1 \wedge f_2 \\ \alpha + max(minTime(f_1), minTime(f_2)) & \text{se } \phi = f_1 \mathbf{U}^\alpha f_2 \\ \alpha + minTime(f_1) & \text{se } \phi = \Box^\alpha f_1 \\ \alpha + minTime(f_1) & \text{se } \phi = \Diamond^\alpha f_1 \end{cases}$$

Definizione semantica

Sia μ una traccia di simulazione con dominio $[s, h]$, ϕ una formula *BLTL* e $t \in [s, h - \minTime(\phi)]$, definiamo allora (in base alla sintassi di ϕ) la validità di $\mu, t \models \phi$:

- **Caso in cui $\phi = \rho$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se il predicato ρ è valido per l'istante t .
- **Caso in cui $\phi = \neg f_1$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se $\mu, t \not\models f_1$ è valida.
- **Caso in cui $\phi = f_1 \vee f_2$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se $\mu, t \models f_1$ e/o $\mu, t \models f_2$ sono valide.
- **Caso in cui $\phi = f_1 \wedge f_2$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se $\mu, t \models f_1$ e $\mu, t \models f_2$ sono valide.
- **Caso in cui $\phi = f_1 \mathbf{U}^\alpha f_2$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se:
 - Esiste un $t' \in [t, t + \alpha]$ per cui $\mu, t' \models f_1$ è valida.
 - Per ogni $t'' \in [t, t')$ vale che $\mu, t'' \models f_2$.
- **Caso in cui $\phi = \Box^\alpha f_1$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se:

Per ogni $t' \in [t, t + \alpha]$ vale che $\mu, t' \models f_1$ è valida.

- **Caso in cui $\phi = \Diamond^\alpha f_1$:**
Diremo che $\mu, t \models \phi$ è valida se e solo se:

Esiste un $t' \in [t, t + \alpha]$ tale che $\mu, t' \models f_1$ è valida.

5 Logica *BLTL**

Consideriamo ora una variante della *BLTL* dove al posto di avere i predicati abbiamo nomi di variabili (o costanti) booleane e chiamato tale logica *BLTL**. Ovviamente la sintassi della *BLTL** è quasi uguale a quella della *BLTL*, l'unica differenza è l'assenza delle relazioni su combinazioni lineari di variabili reali, con al loro posto presenti solo nomi di variabile.

Valutazione La valutazione di una formula *BLTL** richiede una definizione di traccia di simulazione leggermente diversa, il cui codominio non è più l'insieme dei numeri reali \mathbb{R} , ma bensì l'insieme $\{True, False\}$, quindi intuitivamente le variabili non sono più definite su valori reali, bensì su valori booleani. Ciò considerato l'unica differenza

nella semantica consiste nel come è calcolata la valutazione di quelli che nella *BLTL* erano predicati, infatti data una formula contenente soltanto un nome di variabile P , una traccia di simulazione μ ed un istante $t \in [s, h]$, abbiamo che: $\mu, t \models P$ è valida se e solo se $\mu(P, t) = True$

Esempio

Per esempio una formula *BLTL** ϕ potrebbe essere espressa nel seguente modo.

$$\Diamond^{2.03}(P_1 \wedge (True \mathbf{U}^5 P_2))$$

Con appunto P_1 e P_2 nomi di variabile. Assumendo quindi di avere una traccia di simulazione μ definita come:

$$\mu(P_1, t) = \begin{cases} True & \text{se } t \in [0, 5) \\ False & \text{se } t \in [5, 10] \end{cases}$$

$$\mu(P_2, t) = \begin{cases} True & \text{se } t \in [0, 1) \cup [4, 10] \\ False & \text{se } t \in [2, 4) \end{cases}$$

Possiamo affermare che $\mu, 0.5 \models \Diamond^{2.03}(P_1 \wedge (True \mathbf{U}^5 P_2))$ vale, infatti abbiamo che:

1. $\mu, 0.5 \models P_1$ vale banalmente.
2. $\mu, 0.5 \models P_2$ vale banalmente.
3. $\mu, 0.5 \models True \mathbf{U}^5 P_2$ è valido in quanto P_2 è valido per l'istante 0.5.
4. $\mu, 0.5 \models P_1 \wedge (True \mathbf{U}^5 P_2)$ è banalmente valido.
5. $\mu, 0.5 \models \Diamond^{2.03}(P_1 \wedge (True \mathbf{U}^5 P_2))$ è valido in quanto $P_1 \wedge (True \mathbf{U}^5 P_2)$ è valido per l'istante 0.5.