

Descrizione Monitor Generato

Giorgio Mariani

Sommario

In questo documento viene descritto il comportamento di un monitor generato dal sistema.

1 Descrizione monitor

Assumiamo di avere un monitor generato dal sistema a partire da una formula *BLTL* ϕ . Questo monitor consiste in un *subsystem* di *Simulink*[®], composto da differenti tipologie di blocchi interni, in particolare il "nucleo" di questo è dato da una *S-function*. I blocchi in questione sono:

Input Port Blocchi che indicano l'input del sistema, sarà presente un input port per ogni variabile in ϕ , inoltre il nome del blocco sarà lo stesso della variabile. Il blocco ha *data type* scalare reale non complesso.

Output Port Blocco che indica l'output del sistema, questo sarà unico e con *data type* scalare booleano (logical).

Gain Blocchi che prendono in ingresso un segnale e lo

moltiplicano per uno scalare reale costante.

Sum Blocco il cui output consiste nella somma dei segnali in ingresso.

Relational Operator Blocco che prende in input due segnali e restituisce in output un segnale booleano indicante se la relazione associata al blocco è valida o meno.

MUX Blocco che prende in input diversi segnali e li "raggruppa" in un unico vettore

re con larghezza la somma delle larghezze di questi. Per i nostri scopi il MUX prenderà in input n segnali di larghezza 1.

S-function Blocco che consente l'esecuzione, durante una simulazione *Simulink*[®], di una funzione *mex* che rispetti una certa segnatura. Nel monitor sarà presente un blocco *S-function* che esegue un file *mex* in grado di valutare una formula *BLTL**.

Più avanti vedremo come questi blocchi interagiscono tra di loro nel monitor generato.

2 Input del monitor

Gli input del monitor dipendono dall'insieme di variabili presenti in ϕ , infatti per ogni variabile v in ϕ deve essere presente nel monitor una *Input Port* con nome v , tipo di dato **double** e che accetta solo segnali scalari (ovvero con dimensione e larghezza pari a uno) non complessi.

3 Output del monitor

Il monitor deve essere dotato di un'unica porta di output (*Output Port*), con *data type logical* scalare. Il segnale in output del monitor deve rimanere costante tra *Major-timestep*.

Assumiamo di trovarci ad un certo *Major-timestep* di simulazione, allora il segnale in output del monitor assume il valore **1** (*true*) se e solo se la formula ϕ risulta valida per almeno uno tra gli istanti compresi tra l'inizio di simulazione e l'istante associato al timestep corrente meno $\text{minTime}(\phi)$.

nota: Utilizziamo come convenzione che intervalli del tipo $[a, b)$, con $a \geq b$, rappresentino l'insieme vuoto.

Descrizione formale output

Una descrizione più formale del valore restituito in output del monitor durante un certo *Major-timestep* mt potrebbe essere la seguente.

Siano:

- t_s l'istante iniziale di simulazione.
- t l'istante associato al timestep mt .
- μ la traccia di simulazione contenente i valori delle variabili in input al modello fino all'istante t .

Allora l'output del monitor al timestep mt è:

true (1) se e solo se vale che:

$$\exists t' \in [ts, t - \minTime(\phi)) \quad \mu, t' \not\models \phi$$

false (0) se e solo se vale che:

$$\forall t' \in [ts, t - \minTime(\phi)) \quad \mu, t' \models \phi$$

4 Inserimento dei blocchi nel monitor

La costruzione del monitor è organizzata principalmente in tre grossi passaggi:

1. Inserimento dei blocchi che implementano i predicati.
2. Inserimento e collegamento del blocco MUX.
3. Inserimento e impostazione dei parametri della *S-function*.

Costruzione dei predicati Solo i predicati del tipo $\sum \alpha_i x_i \mathcal{R} \beta$ vengono tradotti in blocchi, i predicati del tipo *True* o *False* vengono ignorati. Consideriamo un predicato $\rho = \sum_{i=1}^n \alpha_i x_i \mathcal{R} \beta$, questo viene implementato in più passaggi: Prima viene aggiunto per ogni variabile x_i un blocco *Gain* con con valore di gain α_i e collegato alla *Input Port* con nome x_i . Successivamente i segnali in uscita ai blocchi appena aggiunti vengono dati in input ad un blocco *Sum*, il cui output rappresenta appunto la somma dei segnali in input. Infine vengono aggiunti un blocco con valore costante β ed un blocco *Relational Operator* collegato alla costante appena creata e al segnale in output del blocco somma. Il *Relational Operator* deve avere impostata come relazione il valore di \mathcal{R} .

Inserimento MUX ed S-function Dopo che tutti i predicati sono stati implementati viene costruito un *MUX*, il cui input consiste nei segnali booleani rappresentati i predicati (ovvero gli output dei *Relational Operator*) e l'output consiste in un vettore contenente tutti i segnali in input. Viene inserita infine una *S-function*, prendente in ingresso tale vettore e avente come parametro una conversione a *BLTL** della formula ϕ , dove ogni predicato è identificato da un indice (utilizzato per ottenere il valore del predicato nel vettore in input alla funzione). L'output della *S-function* viene collegato alla *Output Port*.